NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A MODAL ANALYSIS OF CURRENTS IN A PACIFIC ATOLL LAGOON

by

Burton T. Palmer

June 1994

Thesis Advisor: Everett Carter

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June, 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE A MODAL ANALYSIS OF CURRENTS IN A PACIFIC ATOLL LAGOON	5. FUNDING NUMBERS
6. AUTHOR(S) Burton T. Palmer	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE A

13. ABSTRACT *(maximum 200 words)*

The results of an initial quantitative study to define the physical oceanography of Johnston Atoll's lagoon circulation is presented. A bathymetric data base of the Island has been manually digitized from Navigational charts. A numerical model designed to predict the seiche modes of basins in two and three dimensions, in elliptic and Cartesian coordinates is developed, which seems to represent accurately the free modes of oscillation observed in current meter records. The numerical model is suitable for applications using a personal computer equipped with MATLAB (4.0) or later, and has applicability to basins of arbitrary shape in two dimensions, and rectangular, elliptical or circular symmetry in three dimensions. An analysis of currents within the lagoon at Johnston Atoll shows highly polarized tidal flow, phase locked to the diurnal tide. Spectral energy content of current meter records show six fundamental oscillatory modes and harmonics of tidal components.

14. SUBJECT TERMS Johnston Atoll, Seiche, Circulation,Modal Analysis, Oscillation Numerical			15. NUMBER OF PAGES 176 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

Approved for public release: distribution is unlimited.

A Modal Analysis of Currents in a Pacific Atoll Lagoon

by

Burton T. Palmer
Lieutenant Commander, United States Navy
B.S., California State University, Fullerton, 1982

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN Physical Oceanography

from the

NAVAL POSTGRADUATE SCHOOL
June 1994

ABSTRACT

The results of an initial quantitative study to define the physical oceanography of Johnston Atoll's lagoon circulation is presented. A bathymetric data base of the Island has been manually digitized from Navigational charts. A numerical model designed to predict the seiche modes of basins in two and three dimensions, in elliptic and Cartesian coordinates is developed, which seem to represent accurately the free modes of oscillation observed in current meter records. The numerical model is suitable for applications using a personal computer equipped with MATLAB (4.0) or later, and has applicability to basins of arbitrary shape in two dimensions, and rectangular, elliptical or circular symmetry in three dimensions. An analysis of currents within the lagoon at Johnston Atoll shows highly polarized tidal flow, phase locked to the diurnal tide. Spectral energy content of current meter records show six fundamental oscillatory modes and harmonics of tidal components.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. JOHNSTON ISLAND - A GENERAL OVERVIEW 2

B. BATHYMETRY 3

C. PHYSICAL CHARACTERISTICS OF THE LAGOON 5

D. FREQUENCIES OF INTEREST 9

II. MODELING OF JOHNSTON ATOLL OSCILLATORY MOTION 11

A. THE PHYSICS OF OSCILLATORY MOTION - SEICHE 11

B. DERIVATION OF THE EQUATIONS REPRESENTING TWO-DIMENSIONAL OSCILLATION 13

C. NUMERICAL SCHEME FOR THE TWO-DIMENSIONAL CASE . 15

D. NUMERICAL SCHEME FOR THE THREE-DIMENSIONAL CASE IN RECTANGULAR COORDINATES 19

E. NUMERICAL SCHEME FOR THE THREE-DIMENSIONAL CASE IN ELLIPTICAL CYLINDRICAL COORDINATES 23

1. Derivation In Elliptical Cylindrical Coordinates 23

2. Elliptic Coordinate Grid Scheme 28

III. NUMERICAL RESULTS OF THE SEICHE MODEL 34

A. TWO-DIMENSIONAL NUMERICAL RESULTS 34

B. THREE-DIMENSIONAL MODEL RESULTS 40

IV. ANALYSIS OF CURRENTS 52

A. DATA AND METHODS 52

1. Current Meter Data Collection 52

B. SPECTRAL ANALYSIS - FREQUENCY DEPENDENCE 54

C. CURRENTS IN GENERAL 61

V. CONCLUSIONS 74

A. SUGGESTIONS FOR FUTURE RESEARCH 75

LIST OF REFERENCES 77

APPENDIX 79

INITIAL DISTRIBUTION LIST 164

LIST OF TABLES

TABLE 1 DERIVED VOLUMETRIC ESTIMATES 7

TABLE 2 MODEL RESULTS OF TWO DIMENSIONAL GEOMETRIC APPROXIMATIONS 50

TABLE 3 MODEL RESULTS OF Three-dimensional RECTANGULAR APPROXIMATIONS 50

TABLE 4 MODEL RESULTS OF THREE DIMMENSIONAL ELLIPTICAL APPROXIMATIONS 51

TABLE 5 CURRENT METER LOCATIONS 54

TABLE 6 MAJOR COMPONENTS OF CURRENT ENERGY 61

TABLE 7 VOLUMETRIC TRANSPORTS AT SELECTED LOCATIONS 72

LIST OF FIGURES

Figure 1 Johnston Atoll 4

Figure 2 Johnston Atoll 6

Figure 3 Johnston Atoll 8

Figure 4 General Elliptical Coordinates 24

Figure 5 Interpolated Bathymetry 30

Figure 6 Geometry at Foci 31

Figure 7 Elliptic Singularity Condition 32

Figure 8 Rectangular Flat Basin 35

Figure 9 Rectangular Sloping Basin 36

Figure 10 Elliptical Basin, Flat 37

Figure 11 Elliptical Sloping Basin 38

Figure 12 Johnston Atoll First 5 2d modes 39

Figure 13 Mode (1,0) T = 138.8 min 42

Figure 14 Mode (1,0) T = 138.8 min 43

Figure 15 Mode (2,0) T = 69.8 min 44

Figure 16 Mode (2,0) T = 69.8 min 45

Figure 17 Mode (0,1) T = 43.46 min 46

Figure 18 Mode (0,1) T = 43.46 min 47

Figure 19 Mode (1,1) T = 34.82 min 48

Figure 20 Mode (1,1) T = 34.82 min 49

Figure 21 Regression lines of Velocities 53

Figure 22 Energy Density of Current Speed 56

Figure 23 Energy Density of Basin Pressure 57

Figure 24 Energy Density of Model Generated Tide 58

Figure 25 Energy Density of Currents at Station L22 60

Figure 26 U, V, and Pressure at Station MN4 62

Figure 27 Compass plots of Selected Locations 64

Figure 28 Moorings L22 and M14 65

Figure 29 Mooring B14 66

Figure 30 Mooring MN4 67

Figure 31 Moorings HEB, PL1, PL2 68

Figure 32 Moorings PLH, PL3 69

Figure 33 Moorings WCH, WE5, CG4 70

ACKNOWLEDGMENTS

I would like to offer my sincere appreciation to the many people who have provided moral and scientific support during my pursuit of a Master's Degree. I would first like to thank my advisors, Professors Everett Carter and Edward Thornton, for their guidance, patience, wisdom and encouragement that kept me on track. I would also like to thank my curriculum advisors, Commander Tim Cummings, USN and Commander Tim Lage, USN whose counselling and assistance kept me going. Next, I would like to thank the United States Navy for allowing me this great educational opportunity. I would also like to express my appreciation to Professor Eugene Haderlie for rekindling my interest in the ocean's ecology, and for reminding me to always remember that planet Earth is not just ours to use, but ours to protect.

Most importantly, I would like to thank my family, my son Daniel, my daughter Kimberly, and especially my wife Deborah, who have survived patiently and endured the many long days and nights. I know this was not your idea of shore duty, and I thank you for your constant love and companionship, for without you none of this would have been possible. I would also like to say a special thank you to my father and mother, Troy and LaDonne Palmer, who have for years provided the foundation of strength to our family and for compelling me to succeed.

I. INTRODUCTION

The purpose of this study is to begin to quantify the physical oceanography of the Lagoon at Johnston Atoll. This is the first step of a much larger effort to determine the ecological impact of the Island's use. In order to begin a systematic examination of the lagoons circulation, the basic bathymetry and lagoon geometry was digitized to a numerical grid. This enabled the calculation of volume estimates, tidal exchange and tidal prism. The next step was to begin a numerical calculation of the lagoon's circulation.

Prior to any numerical modeling of the circulation, the theoretical periods of oscillation must be known. These include tidal, inertial, seiche and wind event scale frequencies and their harmonics and modulations. As the tidal and Inertial frequencies are readily calculated, the seiche, and important modulations or harmonics need to be identified. To accomplish this, a finite difference numerical model of the seiche was developed. It's results were then compared to an analysis of current meter records. General flow patterns were then identified and some volume transports were estimated.

In Chapter II, a model of the theoretical seiche (modes of free oscillation under the restoring force of gravity) is presented. The model results, offered in Chapter III, appear to accurately approximate the manifestation of seiche-produced current energy within the lagoon.

In Chapter IV, current meter records have been analyzed, and basic circulation patterns have been investigated. Conclusions are presented in Chapter V and the Appendix contains documentation and computer code for the seiche model.

A. JOHNSTON ISLAND - A GENERAL OVERVIEW

Johnston Island is a Pacific atoll situated along the path of the North Equatorial Trade Winds, located in the central Pacific at Latitude 16° 44' N, longitude 169° 32' W. The island is a United States possession and the site of a national wildlife refuge. Once the launch and monitoring site of atmospheric nuclear testing, the island remains under US caretaker status. In 1986, construction began on an environmentally sound, bomb disposal plant for chemical munitions. The Johnston Atoll Chemical Agent Demilitarization System (JACADS) is now in operation.

Typical of most Pacific atolls, the island is the remains of an extinct volcano rising from the deep abyssal plain. Extensive coral growth atop a basaltic core, now cover all of the island. A very shallow lagoon is bounded to the northwest by a linear exposed coral reef, which remains awash for much of the tidal cycle. Gaps within the reef, especially at Monson's Gap, allow exchanges with the open sea. On the southeastern edge of the island, the reef is battered continuously by wave and surf action, limiting the coral growth. Located within the lagoon is one main island, three smaller islands and abundant exposed coral outcroppings.

Studies are underway concerned with the biological impact of the Island's use, but very little has been done with respect to the physical oceanography of the atoll. The only published study concerning the physical oceanography of Johnston Atoll in the published literature that was found, is Barcley (1972), who attempted to quantify observations of the island's wake effect and draw some conclusions about the island's effect on the greater circulation. Nelson (1993), attempted to make indirect measurements of currents over an off shore reef using time domain and frequency domain parameters and simple nonlinear wave solutions. Pickard et.al. (1989) reviews the physical oceanography of the Australian Great Barrier Reef, and determined water properties are closely correlated with air temperatures. Pickard and Emery (1990) discuss Pacific Atolls and reefs in general terms. This study is the first step in quantifying the physical properties of the Johnston Atoll interior lagoon circulation.

B. BATHYMETRY

Detailed hydrographic survey bathymetry was not available at the outset of this project. A digital data set was created from NOAA Navigational chart 83637. which lists soundings in fathoms referenced to mean lower low water (MLLW). Manually digitized to an (x,y) grid of 250 meter resolution, depths were converted to meters. From this data set a variety of subsets were extracted within the lagoon, consisting of depths up to and including the 20 meter isobath. Figure 1, is a three dimensional view of the Bathymetry. Note the steepness of the atoll

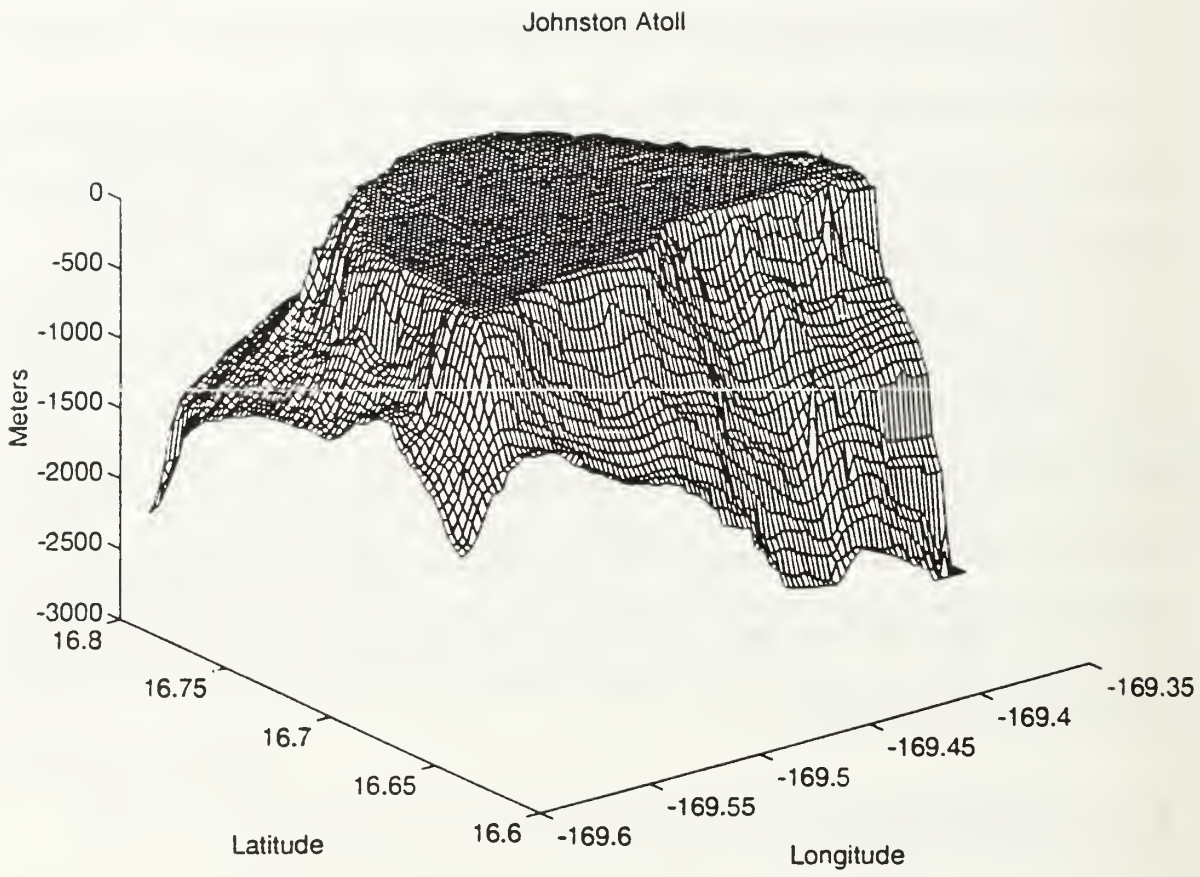


FIGURE 1. Three dimensional rendition of manually digitized bathymetry. Resolution is 250 m.

bathymetry as the depths exceed 20 meters. Outside the 20 meter isobath, the contours abruptly steepen to a gradient averaging 1:1 where the sea floor reaches a depth of over 2800 meters. It was for this reason, that 20 meters was selected as the limits of the lagoon. The limits of the lagoon were chosen from either of two criteria:

1. The northwestern boundary was chosen as the arc containing the exposed reef along the northwest edge.

2. The south and eastern boundary was equated to the twenty meter isobath.

Outside of the reef, the bathymetry drops with near vertical slope to the deep sea floor. Figure 2, is a contoured rendition of the atoll from the digitized data set.

C. PHYSICAL CHARACTERISTICS OF THE LAGOON

The basic geometry of the lagoon suggests an elliptical coordinate system. The lagoon has a maximum length of approximately 16.5 km, and a maximum width approaching 9 km, and a mean depth of about 9.7 meters referenced to mean sea level. A series of dredged shipping channels alter the natural bathymetric profile. Navigational charted tidal records NOAA (1965), and model generated tidal amplitudes, Micronautics (1994), indicate a long term mean tidal excursion of about 1 meter, (mean higher high water (MHHW) - mean lower low water (MLLW)). Calculations of bay volume between the two means give a first

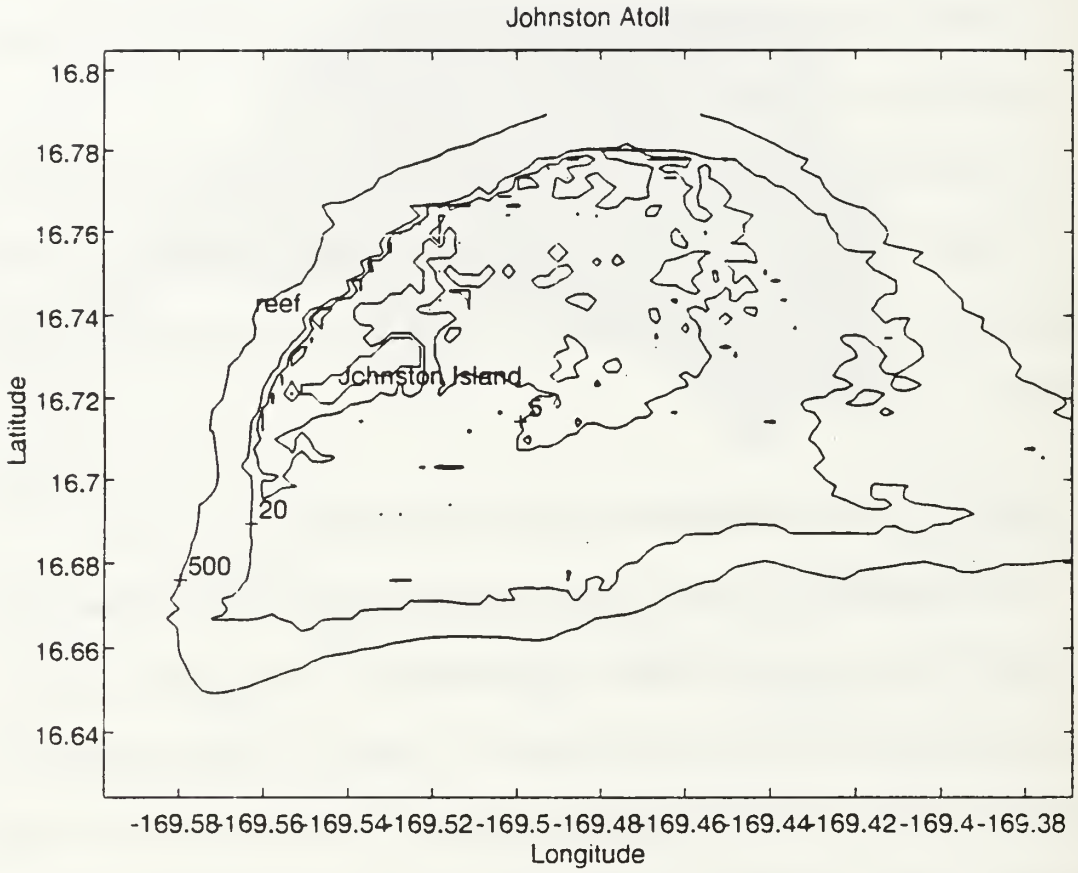


FIGURE 2. Contours of Bathymetry in meters. Resolution is 250 meters.

order estimate of the tidal exchange, and tidal prism. As the sides of the lagoon are leaky, it is difficult to determine the specifics about the tidal exchange. Table 1 lists the basic volumetric estimates derived from the digitized bathymetry.

TABLE 1 DERIVED VOLUMETRIC ESTIMATES.

Surface area	1.143×10^{08}	square meters
Volume		
MHHW	1.130×10^{09}	cubic meters
MLLW	1.016×10^{09}	cubic meters
Tidal Prism	1.15×10^{08}	cubic meters
Mean Depth		
MHHW	10.2	meters
MLLW	9.2	meters

The arc of the reef forms a natural barrier and is well approximated by an elliptical arc. The southern boundary is less well approximated by an ellipse, however, for the discussion presented here, it serves the purpose well. Orthogonal axes across the maximum island dimensions from the northwest reef to the 20 meter isobath along the south and east, reference the coordinate system used in this analysis. Figure 3, shows the geometric layout and coordinate system.

Having established the physical geometry of the lagoon and its volumetric tidal flow, we can estimate, in a simplistic fashion, the mean residence time of a water particle within the lagoon. By making the assumption, that for a given tidal cycle, water enters the lagoon, mixes completely with the existing water, then exits the lagoon via the down stream flow never to return, roughly 10 percent of

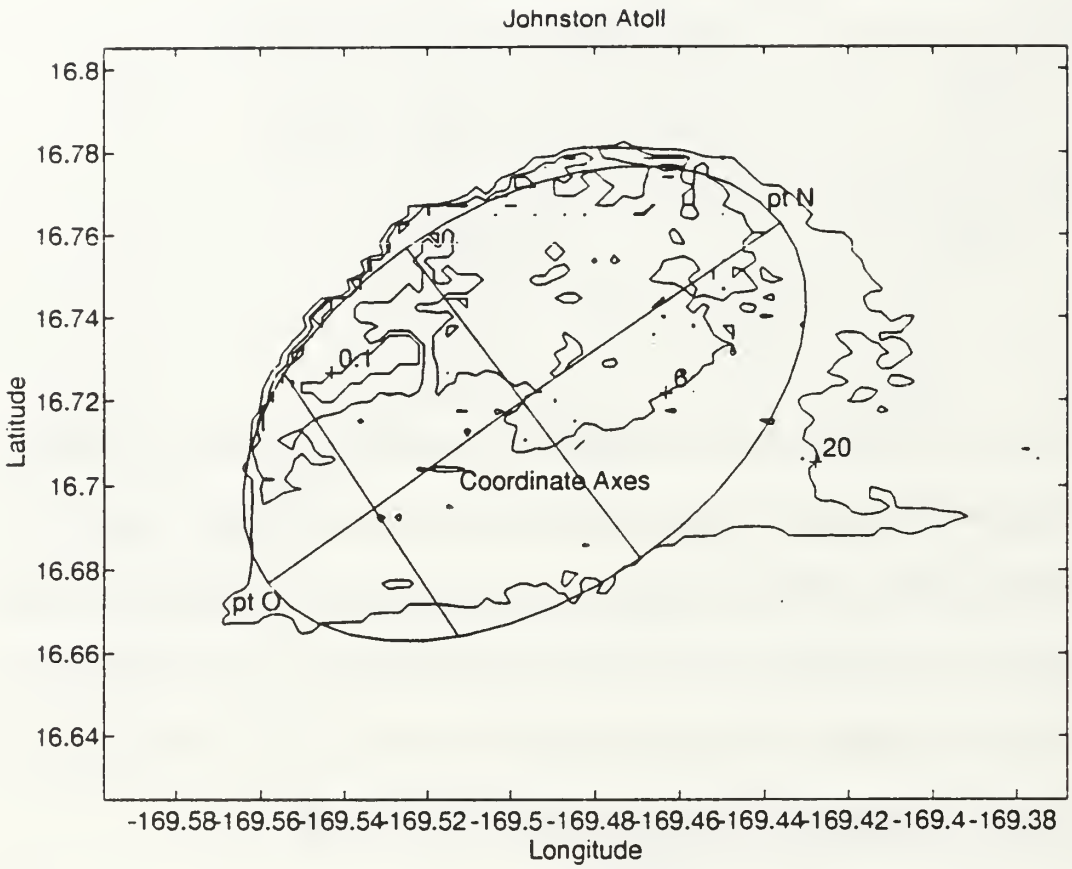


FIGURE 3. Elliptical approximation used in the model overlaid on bathymetric contours. The orthogonal axes are shown. Contours in meters. Points 0 and N represent the axis of the thalweg used in the two-dimensional model.

the lagoon's volume is replaced each 24 hours. This, in a grossly simplified manner, gives an estimate of the residence time of approximately 10 days. A thourough examination must account for the physical processess involved in the true nature of exchanges between exterior flow and the lagoon, evaporation, precipitation, etcetera. Rigorous calculations by Pickard (1990) indicates residence times for a variety of Pacific atoll lagoons to be between two and 28 days. This ten day estimate, therefor, seems reasonable.

D. FREQUENCIES OF INTEREST

The primary frequencies believed to be important in the lagoon's circulation are the diurnal and semi diurnal tidal frequencies, inertial frequency, and the fundamental period of free oscillation. Modulations and/or harmonics of these frequencies may also manifest themselves in current meter observations. Tidal constituents of primary importance are the principal lunar diurnal (M2) and principal semi-diurnal (K1) having periods of 25.3 and 12.2 hours respectively. The inertial period is readily calculated as

$$\frac{1}{f} = \frac{\pi}{\Omega \sin(\phi)} \qquad (1)$$

where ϕ is latitude and Ω is the angular velocity of terrestrial rotation and equal to 7.292×10^{-5} (rad/s). Given the island's latitude of 16.6^0 N, the inertial period is 41.56 hours.

An accurate estimate of the seiche is required, to further quantify the tidal exchanges, flow patterns, and to begin an examination of the circulation. Observations of currents within the lagoon should identify whether seiche is important in the lagoon's circulation.

II. MODELING OF JOHNSTON ATOLL OSCILLATORY MOTION

The seiche modes are examined with the development of three numerical models. The first involves 2 Cartesian dimensions (length and depth (x,z)); the second involves three Cartesian dimensions (length, width, and depth: x, y, z); the third involves 3 dimensions in elliptical cylindrical coordinates,(υ, μ, z). These models are programmed in MATLAB (4.0) and suitable for use on a personal computer or work station equipped with MATLAB (4.0) or later. All of the models are written general enough to be applied to a variety of geometries. It is hoped the models will be useful in applications and as a teaching aid for analysis of bays and harbors other than just the Johnston Atoll.

A. THE PHYSICS OF OSCILLATORY MOTION - SEICHE

If a free wave propagating in shallow water is constrained within a basin, and travels at a speed $c= \sqrt{(gh)} = \lambda/T$, g being gravity, h being water depth, λ being wavelength and T the period, then Defant (1960) showed that the relation $L = \lambda/2$ is the maximum standing wave possible within the confines of an enclosed basin of length L. In an open mouthed basin, $L = \lambda / 4$ limits the wavelength of a standing wave. The physical characteristics of such a wave are totally dependent on the geometry of the basin. External forcing by winds, tides or inertial oscillations that coincide with the fundamental period of oscillation within a basin, can induce resonance. The most widely known resonance is perhaps the Bay of Fundy, where

the fundamental period of oscillation is very close to the semi-diurnal tidal period. In this extreme example, huge tidal ranges are experienced as the two mechanisms constructively interfere. The resonant oscillation of bays and harbors has been fairly well compiled. Defant (1960) studied the effect in detail and developed analytical solutions to the fundamental equations that approximate the phenomena for many bays and harbors. Goldsbrough (1930) studied tidal oscillations, in a relatively sophisticated way for his time, using an elliptic approximation for a resonant harbor.

Seiche is the chain type resonance that results when a resonator is placed inside another resonator. Chain resonance results when the frequencies of the innermost resonator are also resonant frequencies of the outer resonators. In the next section, the steady state oscillatory nature of the Johnston Atoll lagoon is developed and analyzed. Wilson (1965) serves as the basis of this model. His study of Monterey Bay well summarizes the physics for open mouthed bays. In attempting to numerically solve for the seiche modes, however, the computing capabilities of the time, and some minor numerical errors in forming the boundary conditions limited Wilson's success in predicting seiche modes in three dimensions. An adaptation of the finite difference scheme applied in Wilson (1965) is presented here. First, a two dimensional model is developed. Next, the model is expanded into three dimensions in rectangular coordinates. The rectangular system allows computational and programming simplicity, illustrates the fundamental concepts, and should apply to bays of near rectangular geometry. It

is, however, incapable of representing a curved boundary as found at Johnston Atoll. For this reason, the three-dimensional model is altered to reflect an elliptical cylindrical coordinate system and applied to the geometry of the Johnston Atoll.

B. DERIVATION OF THE EQUATIONS REPRESENTING TWO-DIMENSIONAL OSCILLATION

This derivation closely follows that of Wilson (1965). The general equations of motion and continuity are in the form represented by Defant (1960). For a semi enclosed bay, oscillatory waves in rectilinear coordinates can be expressed in the form:

$$(A(x)\Phi_x)_x = \frac{b(x)}{g}\Phi_{tt} \qquad (2)$$

where A(x) is crossectional area, at points along the x axis, taken to be the thalweg of the basin, Φ is a velocity potential and g is gravity. Equation 1 must satisfy the linearized free surface condition represented by:

$$\Phi_t = -g\eta(x) \qquad (3)$$

where η is the perturbed free surface elevation measured with respect to the mean. Assuming a separable form of $\eta(x,t)$;

$$\eta(x,t) = \eta(x)e^{i\sigma t} \tag{4}$$

where σ is the frequency of oscillation,

$$\sigma = \frac{2\pi}{T} \tag{5}$$

and T is the period of a surface wave. Substituting for η, equation (2) becomes:

$$(A(x)\eta_x)_x + \frac{\sigma^2 b(x)\eta}{g} = 0 \tag{6}$$

which becomes the equation describing the behavior of the surface elevation. Letting A = b(x)z(x) (width x depth), and carrying through the indicated differentiation, equation (6) becomes:

$$bz\eta_{xx} + (bz_x + zb_x)\eta_x + \frac{\sigma^2 b\eta}{g} = 0 \tag{7}$$

This is the general, two dimensional equation governing oscillations in an open basin. Solutions of this equation describe the surface behavior once an appropriate boundary condition is formulated. At point (0) a node line is assumed. For the Johnston atoll this is assumed to coincide with the 20 meter isobath. Here,

$\eta(0)=0$. At point N, the enclosed boundary, two boundary conditions on η were considered. These are determined by the bathymetry at the boundary $\eta(N)$. Boundary condition 0: if z approaches 0 at the boundary, this implies $\eta = 0$ at the boundary, and equation (7) can be simplified to

$$z_x \eta_x \mid_N + \frac{\sigma \eta_N}{g} = 0 \qquad (8)$$

Boundary condition 1: if z is of finite depth at the boundary, the velocity normal must equal zero which means $\eta_x \mid_N = 0$, and equation (7) becomes:

$$z(\eta_{xx}) \mid_N + \frac{\sigma \eta}{g} = 0 \qquad (9)$$

Either of the boundary equations may apply for a given basin. Solutions of (7) are the solutions of an eigenvalue problem, where x represents the set of all eigenvectors (modes) and λ represents the eigenvalues associated with each eigenvector.

C. NUMERICAL SCHEME FOR THE TWO-DIMENSIONAL CASE

Taking the thalweg of the lagoon as the x axis and averaging the depths (z) along the thalweg at discrete points between 0 and N, finite difference approximations at each point as given by Wilson (1965) were converted to numerical form after correction of some minor errors. Equations (7), (8), and (9)

were nondimensionalised in the usual manner. Nondimensionalization involves letting

$$\bar{z}_i = \frac{z_i}{z_o} \quad , \quad \bar{b} = \frac{b_i}{L} \quad , \quad \Delta x = \frac{L}{N} \tag{10}$$

where z_o = depth at point O, L = length of basin and N is number of points considered and the subscript i indicates the index of points 1 to N. Upon substituting (10) into (7) and dropping the overbars, the three point centered difference form of equation(7) is given by Wilson(1965) as:

$$\begin{aligned} &\eta[2z_i - \lambda] - \eta_{i+1}\{z + \frac{1}{4}[z_{i+1} - z_{i-1} + (\frac{z_i}{b_i})(b_{i+1} - b_{i-1})]\} \\ &- \eta_{i-1}\{z_i - \frac{1}{4}[z_{i+1} - z_{i-1} + (\frac{z_i}{b_i})(b_{i+1} - b_{i-1})]\} = 0 \end{aligned} \tag{11}$$

where

$$\lambda = \frac{\sigma^2 L^2}{N^2 g z_o} \tag{12}$$

is the eigenvalue. Equation (11) applies at points i = 1 through N-1. Equation (8) or (9) apply at the boundary i = N, providing a complete set of equations.

Equations (8) and (9) must be differenced using a three point backwards technique. The numerical equivalent of (8) is

$$\eta_N \{ \frac{3}{4} [4z_{N-1} - z_{N-2}] - \lambda^2 \} - \eta_{N-1} (4z_{N-1} + z_{N-2}) + \frac{1}{4} \eta_{N-2}(4z_{N-1} - z_{N-2}) = 0 \quad (13)$$

and the numerical equivalent of (9) is:

$$\eta_n(2z_n) - \eta_{N-1}(5z_N) + \eta_{N-2}(4z_N) - \eta_{N-3}(z_N) = \lambda^2 \quad (14)$$

For numerical solutions to the problem of N points, N-1 equations of the type (11) and one equation of the type (13) or (14) are formed. This results in an eigenvalue matrix of size N x N. Solving this eigenvalue matrix yields the eigenvalues (λ_i) which contain the periods of oscillation (T_i). The corresponding eigenvectors (η_i) represent the mode shapes of the oscillation.

Numerical eigenvalue solvers are readily available. The one chosen for this model is the standard eigenvalue solver supplied with the numerical package MATLAB (4.0), (1993). This algorithm uses EISPACK routines to solve for the generalized eigenvalues and eigen vectors via the QZ method. Details for the algorithm are found in the EISPACK guide, Garbow, (1977).

Upon verification of the model scheme with the Monterey Bay geometries presented in Wilson (1965), and retrieving essentially the same results, the method was then applied to the approximations of the Johnston Atoll lagoon. The first two-dimensional approximation was a square basin with a flat bottom of depth 9.5 meters. Next an elliptical approximation was made to the lagoon boundary, and finally the real bathymetry and geometry was applied. The analytic solution of equation 1 in two dimensions is also presented for the uniform depth, constant width case. It can be shown that the analytic solution in two dimensions of equation (2) for a closed basin is:

$$T_n = \frac{2L}{n\sqrt{gh}} \tag{15}$$

and for an open mouthed bay,

$$T_n = \frac{4}{n\sqrt{gH}} \tag{16}$$

Ippen (1966). Here the subscript n represents the modal number. The most appropriate analytical solution for Johnston Island is the open mouthed condition. Equation (16) serves as a baseline to compare against the numerical results in two dimensions.

D. NUMERICAL SCHEME FOR THE THREE-DIMENSIONAL CASE IN RECTANGULAR COORDINATES

Expanding equation (2) in rectangular coordinates is straightforward. By allowing depth to vary in both x and y, equation (2) becomes

$$(z(x,y)\Phi_x)_x + (z(x,y)\Phi_y)_y - \frac{1}{g}\Phi_{tt} = 0 \qquad (17)$$

where Φ is still a potential function evaluated at the equilibrium position of the free surface. The linearized Bernoulli equation of the free surface is:

$$\eta(x,y) = -\frac{1}{g}\Phi_t \qquad (18)$$

As in equation (3), η is the surface displacement from equilibrium. For harmonic motion, the relation

$$\eta(x,y,t) = \eta(x,y)e^{i\sigma t} \qquad (19)$$

can be used. Substituting (19) into (17), the following equation governs the displacement of the free surface.

$$(z\eta_x)_x + (z\eta_y)_y + \frac{\sigma}{g}\eta = 0 \qquad (20)$$

Carrying through the differentiation indicated, and rearranging terms, equation (20) is simply,

$$z_x\eta_x + z\eta_{xx} + z_y\eta_y + z\eta_{xx} + \frac{\sigma^2}{g}\eta = 0 \qquad (21)$$

The analytic three-dimensional, solution to equation (21), for an open mouthed bay is

$$T_{n,m} = \frac{4}{\sqrt{gH}}\left[\left(\frac{n}{a}\right)^2 + \left(\frac{m}{b}\right)^2 \right]^{-1/2} \qquad (22)$$

where n and m represent the modal numbers in the x or y directions, and where a and b represent the length and width of the basin, Ippen (1966). The indices n, m indicate that the oscillation is allowed to propagate in both x and y. T is represented by n,m indices pairs symbolizing the mode of oscillation in orthogonal directions. The indices also represent the number of zero crossings the mode shape takes as the wave oscillates across the length (width) of the basin. Equation (22) serves as a comparative baseline for evaluating the three-dimensional numerical scheme. In a given basin, n and m can be different, resulting in an infinite number of possible combinations. For this reason, results presented here

are simplified to the modes where the combinations of $T_{n,m}$ modes are the ten largest periods.

Letting x = i Δx, and y = j Δy, where i and j are integer indices. (i = 1,2,3 ...N, j = 1,2,3 ... M), the following nondimensionalizations are then applied.

$$z' = \frac{z}{z_o}, \quad \eta' = \frac{\eta}{z_0}, \quad L = Ndx, \quad L' = \frac{L}{L}, \quad W = Mdy, \quad W' = \frac{W}{L} \tag{23}$$

The boundary conditions are similar to the two dimensional case and are applied at the four boundaries. Either of the two following conditions will be true, depending on the bathymetry. Where the water depth (z) is of finite depth at a boundary (x = 0, y = 0, x = L, or y = W), then, the velocity normal to the free surface elevation must equal zero, at that boundary, or

$$(\eta_x)|_I = 0, \quad (\eta_x)|_J = 0 \tag{24}$$

where the I_I and I_J imply evaluation at the boundary points(I = 1 or N, J = 1 or M). Where the water depth approaches zero at a boundary, the free surface elevation must equal zero, and the boundary condition is satisfied with

$$\eta|_{I,J} = 0 \qquad (25)$$

Backward differences of equation 21 or 22 must be applied at I = N, J = M, and forward differences applied at I = 1, J = 1.

After dropping the primes, the three point centered difference form of (21) is

$$\begin{aligned}
&\eta_{i,j}\left[-2\frac{z_{i,j}}{\Delta x^2} - 2\frac{z_{i,j}}{\Delta y^2}\right] + \\
&\eta_{i+1,j}\left[\frac{1}{4\Delta x^2}(z_{i+1,j} - z_{i-1,j} + 4z_{i,j})\right] + \\
&\eta_{i-1,j}\left[\frac{1}{4\Delta x^2}(z_{i-1,j} - z_{i+1,j} + 4z_{i,j})\right] + \\
&\eta_{i,j+1}\left[\frac{1}{4\Delta y^2}(z_{i,j+1} - z_{i,j-1} + 4z_{i,j})\right] + \\
&\eta_{i,j-1}\left[\frac{1}{4\Delta y^2}(z_{i,j-1} - z_{i,j+1} + 4z_{i,j})\right] = \\
&\eta_{i,j}\left[\frac{\sigma^2 L^2}{g z_o}\right]
\end{aligned} \qquad (26)$$

This represents a system of N x M equations. Equation (26) is applied at all the interior points and either of 24 or 25 is applied at the boundaries. The resulting (N x M) x (N x M) coefficient matrix is then solved as an eigenvalue problem using the same technique as in the two dimensional case. As this scheme involves numerical computations proportional to $(N \times M)^2$, computational expense is accumulated as N and M increase.

E. NUMERICAL SCHEME FOR THE THREE-DIMENSIONAL CASE IN ELLIPTICAL CYLINDRICAL COORDINATES

1. Derivation In Elliptical Cylindrical Coordinates

From plane geometry, the basic formula of an ellipse in cartesian coordinates is given as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{27}$$

where a is the semi-major axis and b is the semi-minor axis. Transformation from cartesian coordinates to elliptical cylindrical coordinates follows the relations:

$$x = c_f \cosh\mu \cos\upsilon, \quad y = c_f \sinh\mu \sin\upsilon, \quad z = z \tag{28}$$

where c_f is the center to focus distance, given by

$$c_f = \sqrt{a^2 - b^2} \tag{29}$$

General elliptical cylindrical coordinates are depicted in Figure 4. υ specifies lines of constant angle, $(-\pi \leq \upsilon < 0)$, μ specifies lines of constant curvilinear distance, $\mu \geq 0$. The traces of the coordinate surfaces on the xy plane specify a set of confocal ellipses and hyperbolas. The coordinate surface $\mu = 0$ is a straight line extending between the two foci. The coordinate surface $\mu = 2$ is an ellipse of zero eccentricity, i.e., a circle.

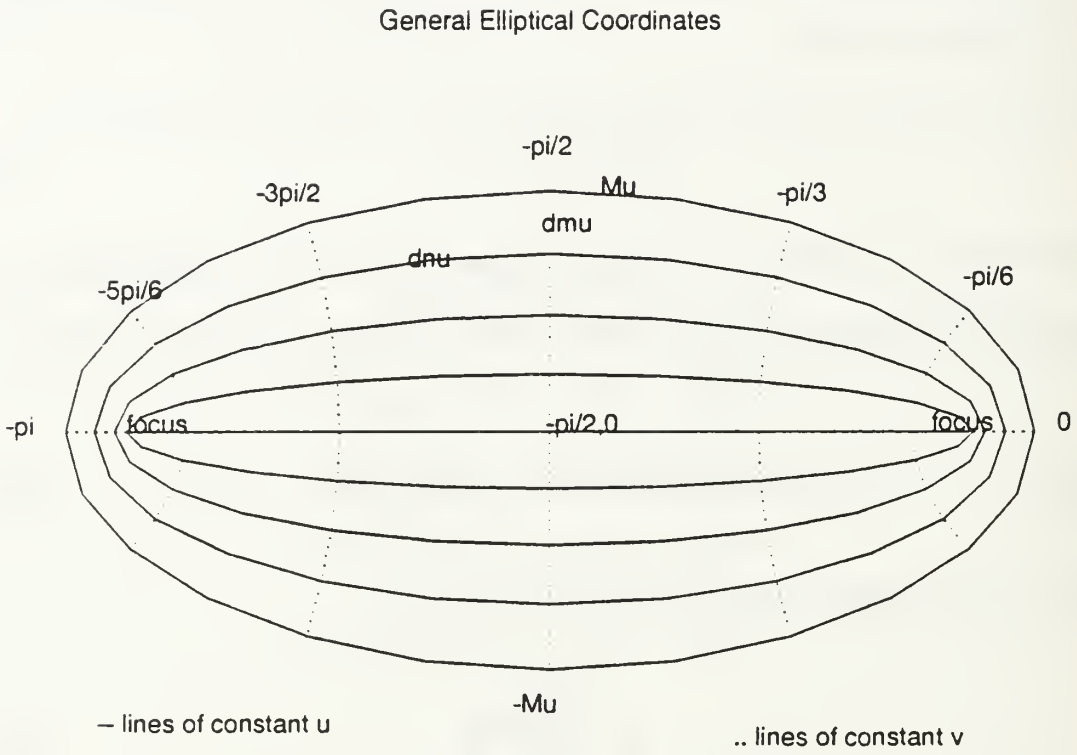


FIGURE 4. Simplified general elliptical coordinate system. At μ = -pi and 0, the coordinates are double specified. See text for amplification.

The general curvilinear coordinate version of equation (21) is given by

$$\frac{1}{h_1 h_2}\left[\frac{h_2}{h_1}(z\eta_{q_1})_{q_1} + \frac{h_1}{h_2}(z\eta_{q_2})_{q_2}\right] + \frac{\sigma^2}{g}\eta = 0 \qquad (30)$$

where h_1, h_2 are the curvilinear distortion factors and q_1, q_2 are the curvilinear coordinates. In elliptical cylindrical coordinates,

$$h_1^2 = h_2^2 = c_f^2(\sinh^2\mu + \sin^2\upsilon) \qquad (31)$$

and, $q_1 = \mu$, $q_2 = \upsilon$. Upon making these substitutions, equation (27) becomes:

$$\frac{1}{h^2}\left[(z\eta_\mu)_\mu + (z\eta_\upsilon)_\upsilon\right] + \frac{\sigma^2}{g}\eta = 0 \qquad (32)$$

Boundary conditions for the elliptical cylindrical coordinate system are the same as for the rectangular case and apply at $\mu = -\mu_{max}$ and $\mu = \mu_{max}$. For the regions where the depth is finite,

$$\frac{1}{h^2_{I,J}}\eta_\mu = 0 \qquad (33)$$

and for the regions where the depth approaches 0,

$$\eta = 0 \qquad (34)$$

Nondimensionalization similarly follows from the rectangular case.

Letting depth and sea surface elevation be scaled by Z_0, a reference depth, and the length and width be scaled by the center to focus distance,

$$z' = \frac{z}{z_o}, \quad \eta' = \frac{\eta}{z_0} \tag{35}$$

$$h_1^{2'} = h_2^{2'} = \frac{c_f^2}{c_f^2}(\sinh^2\mu + \sin^2 v) \tag{36}$$

After carrying through the differentiation, nondimensionalization, and dropping the primes, then applying three point centered differences , equation (32) becomes

$$\begin{aligned}
&\eta_{i,j}\frac{1}{h^2_{i,j}}\left[-2\frac{z_{i,j}}{\Delta\mu^2} - 2\frac{z_{i,j}}{\Delta v^2}\right] + \\
&\eta_{i+1,j}\left[\frac{1}{h^2_{i,j}\,4\Delta\mu^2}(z_{i+1,j} - z_{i-1,j} + 4z_{i,j})\right] + \\
&\eta_{i-1,j}\left[\frac{1}{h^2_{i,j}\,4\Delta\mu^2}(z_{i-1,j} - z_{i+1,j} + 4z_{i,j})\right] + \\
&\eta_{i,j+1}\left[\frac{1}{h^2_{i,j}\,4\Delta v^2}(z_{i,j+1} - z_{i,j-1} + 4z_{i,j})\right] + \\
&\eta_{i,j-1}\left[\frac{1}{h^2 j\,4\Delta v^2}(z_{i,j-1} - z_{i,j+1} + 4z_{i,j})\right] = \\
&\eta_{i,j}\left[\frac{\sigma^2 c_f^2}{g z_o}\right]
\end{aligned} \tag{37}$$

where $\mu = i\,\Delta\mu$ and $\upsilon = j\,\Delta\upsilon$ and i, j are integers: (i = 1,2,3, ... IM), (j = 1,2,3, ...JM). Three point backwards differences of equation 33 yields

$$\eta_N \frac{3}{2\Delta\mu h^2_{i,j}} - \eta_{N-1} \frac{2}{\Delta\mu h^2_{i,j}} + \eta_{N-2} \frac{1}{2\Delta\mu h^2_{i,j}} = 0 \qquad (38)$$

and is applied along the boundary where $z_{1,j}$ or $z_{IM,j}$ is of finite depth. At the boundary, where $z_{1,j} \rightarrow 0$ the condition $\eta = 0$ is automatically applied.

As the bay contains not only varying bathymetry, but islands, interior boundary conditions must be formulated, which must be applied numerically at the five point stars making up the boundaries, surrounding the island. The condition most appropriate is $\eta = 0$.

As in the rectangular geometry, a system of N x M eigenvalue equations is developed and solved for the free modes of oscillation. The resulting N x M eigenvalues contain the periods of oscillation (frequencies) and the N x M eigenvectors represent the modal shapes. These vectors are then analyzed to determine the nodes and anti-nodes within the boundaries of the lagoon. This information will then be used later for the analysis of observational current meter data and placement of future instrumentation. Since the numerical accuracy diminishes with higher eingenvalues, only the first few are of any relevance.

Also, at extremely fine resolutions, the eigenvalue matrix becomes extremely large. The condition number of the matrix also increases causing

computational instability, limiting the eigenvalue solver's ability to solve large matrices. Robust eigenvalue solvers are available, but computationally expensive. Thus, for the computer resources available, a moderately fine resolution averaging an equivalent Cartesian spacing approaching 750 meters gave the best results.

2. Elliptic Coordinate Grid Scheme

An interpolation of depths into the μ, υ space is shown in Figure 5. In its purest form, elliptical cylindrical coordinates are double specified at points along $\upsilon = -\pi$, and $\upsilon = 0$. Special care must be taken to eliminate this duplication in the numerical computation. Thus, in order to digitize the bathymetry, convert to elliptical cylindrical coordinates and compute the coefficient matrix, the numerical domain had to be tailored to an i, j grid corresponding to those points within the boundaries of the ellipse, without duplication of any coordinate points.

In addition, calculations involving the foci must necessarily be carefully treated as they induce a singularity in calculating the curvilinear distortion factor, h^2. As the grid mesh converges to zero at the foci, the value of h^2 goes to zero. The calculation of :

$$\frac{1}{h^2} = \frac{1}{c_f^2(\sinh^2\mu + \sin^2\nu)} \rightarrow \infty \qquad (39)$$

at the foci. Figure 6 shows the details of the 5 point star at the foci. Figure 7 shows the singularity condition of h^2. Two methods were formulated to resolve the difficulty. One widely accepted method is to use the Cartesian equivalent of

equations (37) which are simply equations (26). The motivation for this technique is that, near the foci, the elliptic grid closely resembles Cartesian coordinates. Applying the Cartesian coordinate equations at the five point star surrounding the foci, should theoretically eliminate the singularity, since h^2 does not appear in the numerical equation. Even at relatively course grid spacing, however, the elliptic coordinates converge and the computed values of Δx and Δy become very small when compared to their elliptic equivalents $\Delta \upsilon$ and $\Delta \mu$, mitigating the numerical advantage of the rectangular approach to the problem. It was determined through extensive trouble shooting of the singularity condition, that the rectangular approximation of the foci was less than optimal. The second approach tended to give more stable numerical results over a wider range of resolutions. This method essentially involves removing the foci from the numerical computation altogether and is equally sucessful and less mathmatically cumbersom. Since h^2 is well defined everywhere but at the foci, and the grid mesh is sufficiently fine there, the elimination of the foci as coordinate points was sufficient to resolve the difficulty. As the duplicated coordinate points and the foci are at the outermost extremes of the mesh and (nearest neighbors), elimination of the numerical instability was acheived by removing the data points associated with the problem indices, then renumbering the grid to account for the "missing" points. The ambiguity is a result of the grid rotation, where the two axes collocate. Both the duplicated coordinate

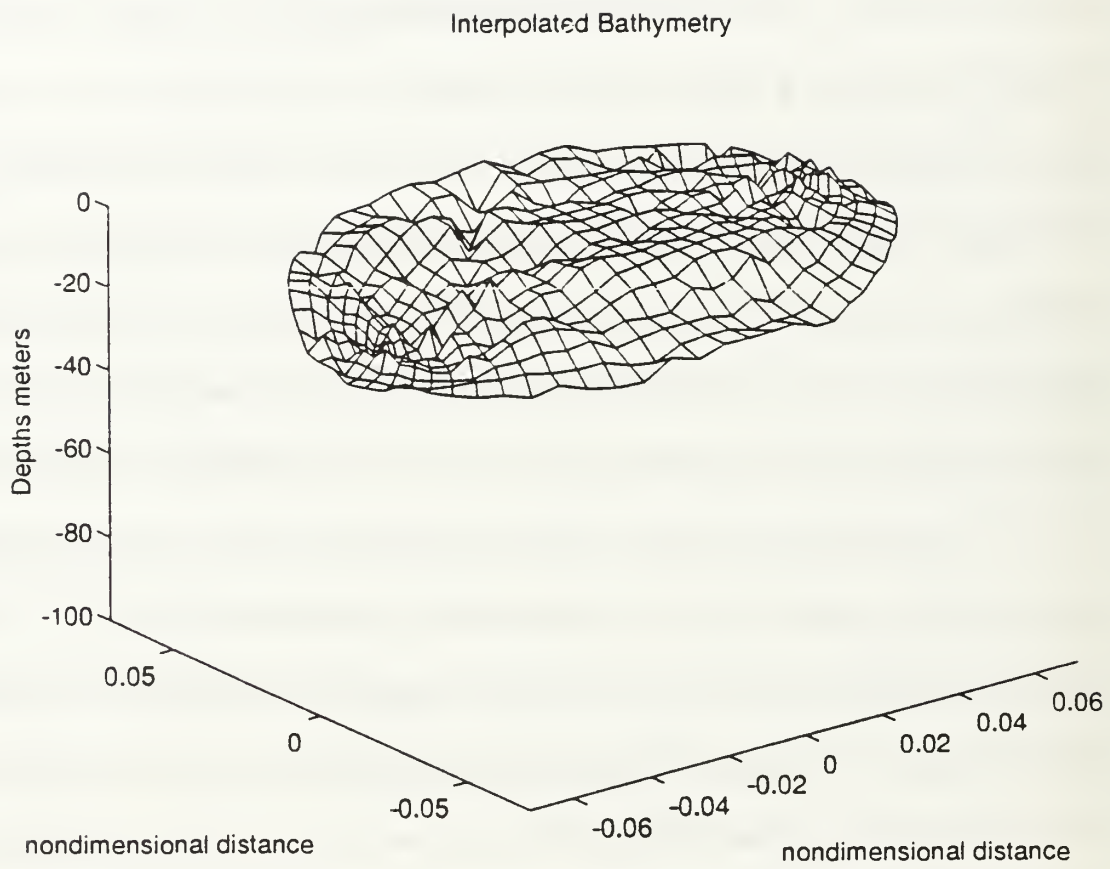


FIGURE 5. Interpolated Bathymetry used in the Three-dimensional seiche model.

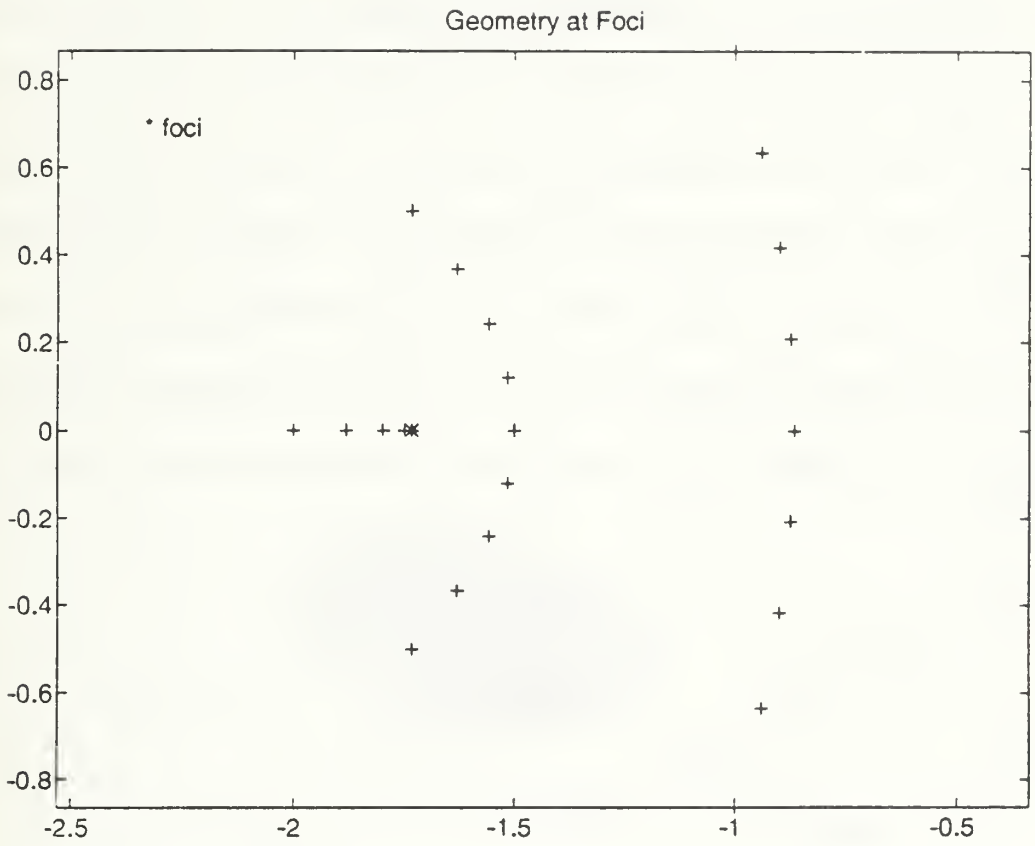


FIGURE 6. Numerical geometry at the foci. Points shown are for computations surrounding the five point star centered at i = 2 , j = (JM+1)/2.

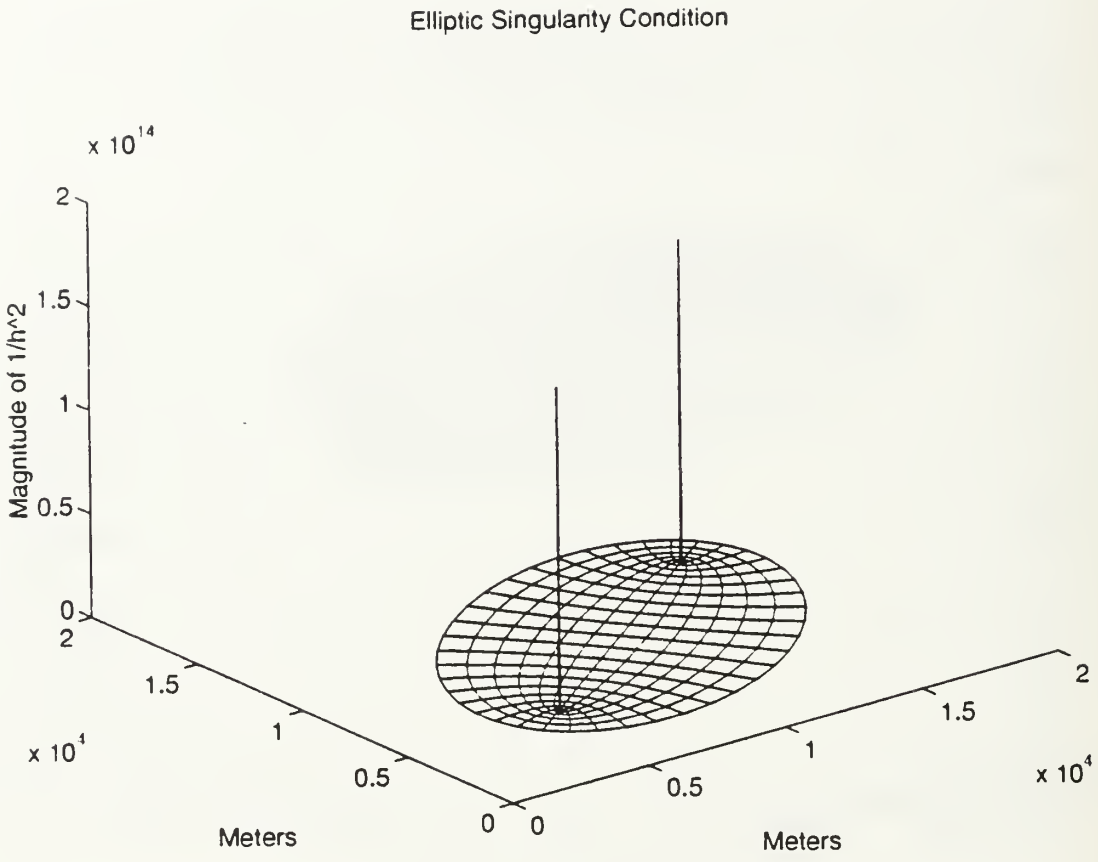


FIGURE 7. Graphical depiction of the singularity condition at the foci. As μ, υ approach the foci, the value of the scale factors approach infinity.

points and the foci were eliminated by deleting the (i = 1, v = $-\pi$), (j = 1 through (JM+1)/2)), and (i = IM, v=0), (j = (JM+1)/2 through JM) indices, then renumbering the remaining points. Proper bookkeeping of nearest neighbor indices in this region required special care. Both versions of the model, (version(f), rectangular foci, and version(g), no foci) , ran successfully, but careful analysis of model results over a range of resolutions determined that the best estimates of the lagoon seiche was achieved with version(g), the foci eliminated. The rectangular coordinates at the foci tended to over estimate the seiche modes by about 20 percent. The results presented in this paper are those of the three-dimensional elliptical model sans foci, (version g). The computer code for all versions is attached in the Appendix.

III. NUMERICAL RESULTS OF THE SEICHE MODEL

A. TWO-DIMENSIONAL NUMERICAL RESULTS

Numerical results at a resolution of Δx = 250 meters for the various model cases are shown in TABLE 2 and Figures 8 through 12. Basin boundaries were chosen, proceeding from simple to more realistic geometric approximations. As the true bathymetry is neither uniformly sloping or flat, the accuracy of the eigenvalue solution should increase as more realistic geometries are applied. The analytic solution is for the rectangular uniform flat basin geometry, L=16,250 meters, width is 6,300 meters. For the elliptical basin, the minor axis width is 8000 meters. The coefficient matrix is of the order 68 x 68, well within the general eigenvalue solver's numerical ability.

The two dimensional model is of limited usefulness as applied to Johnston Atoll. While modal periods converge to reasonable accuracy as bathymetric approximation improves, it gives nodal points as positions only along the thalweg. It is difficult to extrapolate nodal lines away from the thalweg and transfer the geographic positioning of nodal lines to contours of the lagoon's bathymetry. In addition, the openness of the bay's boundary is not considered except at point 0 and N.

For the first 10 modes, the numerical solution of a flat bottom domain closely agrees with the analytic one. The modal periods increase as better

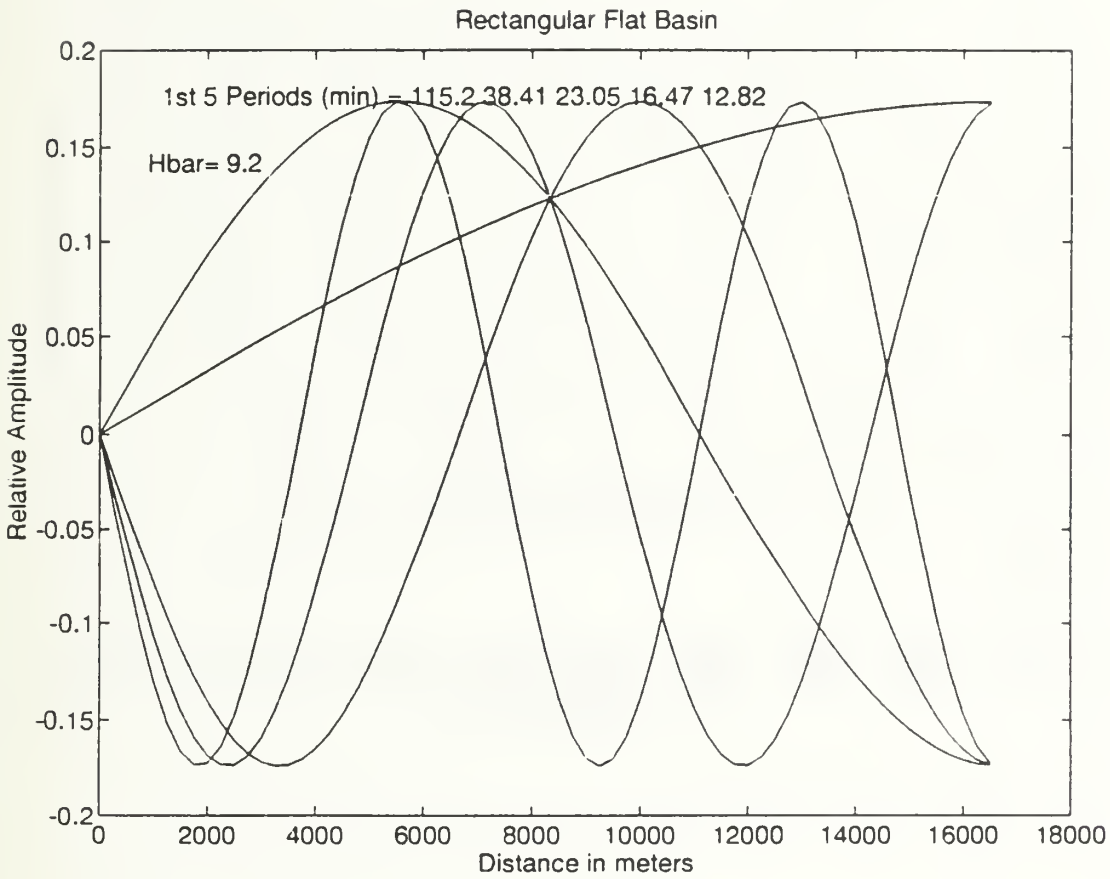


FIGURE 8. Numerically computed sinusoidal shapes of the first 5 modes of seiche for a rectangular approximation to the lagoon with a mean depth of 9.12 meters. L= 16,250 meters, W = 6300 meters.

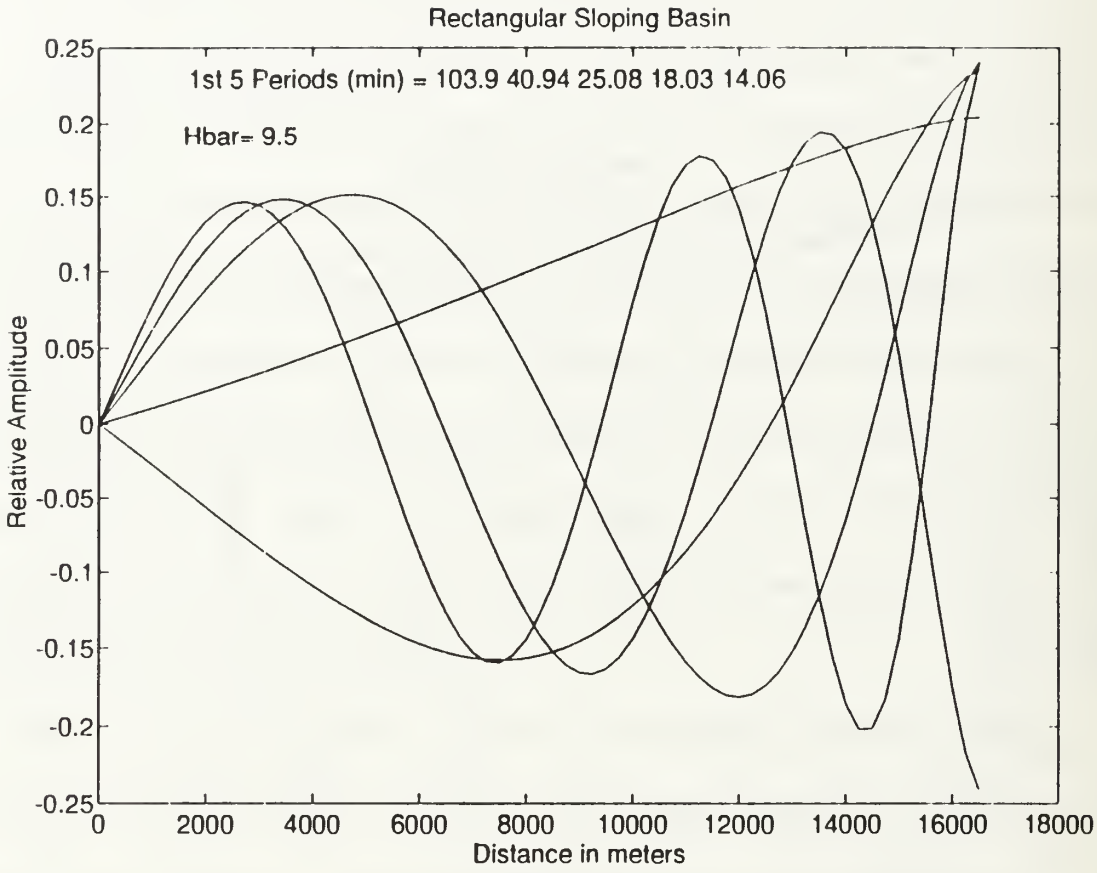


FIGURE 9. Same as figure 8 except using a uniformly sloping bottom from 20 meters to zero.

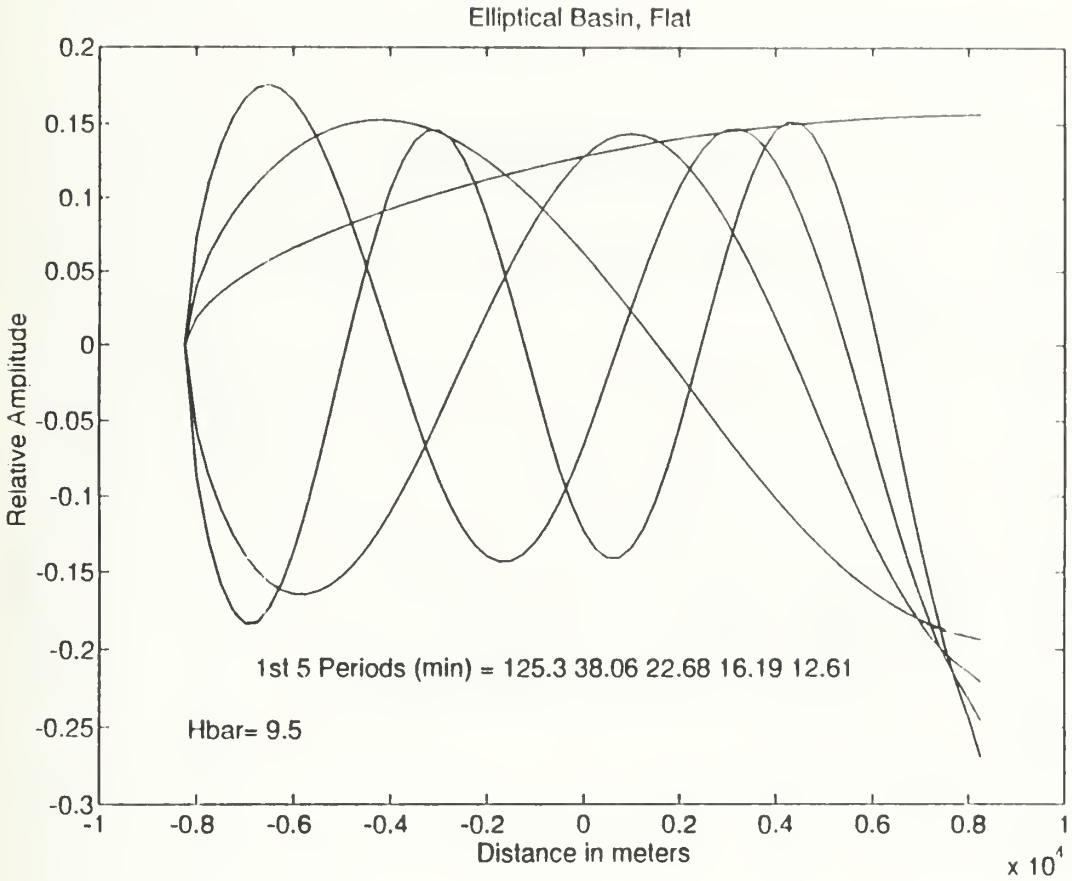


FIGURE 10. Numerical solution of an elliptical basin boundary with a flat bottom. The modal period has increased with improved accuracy of the model.

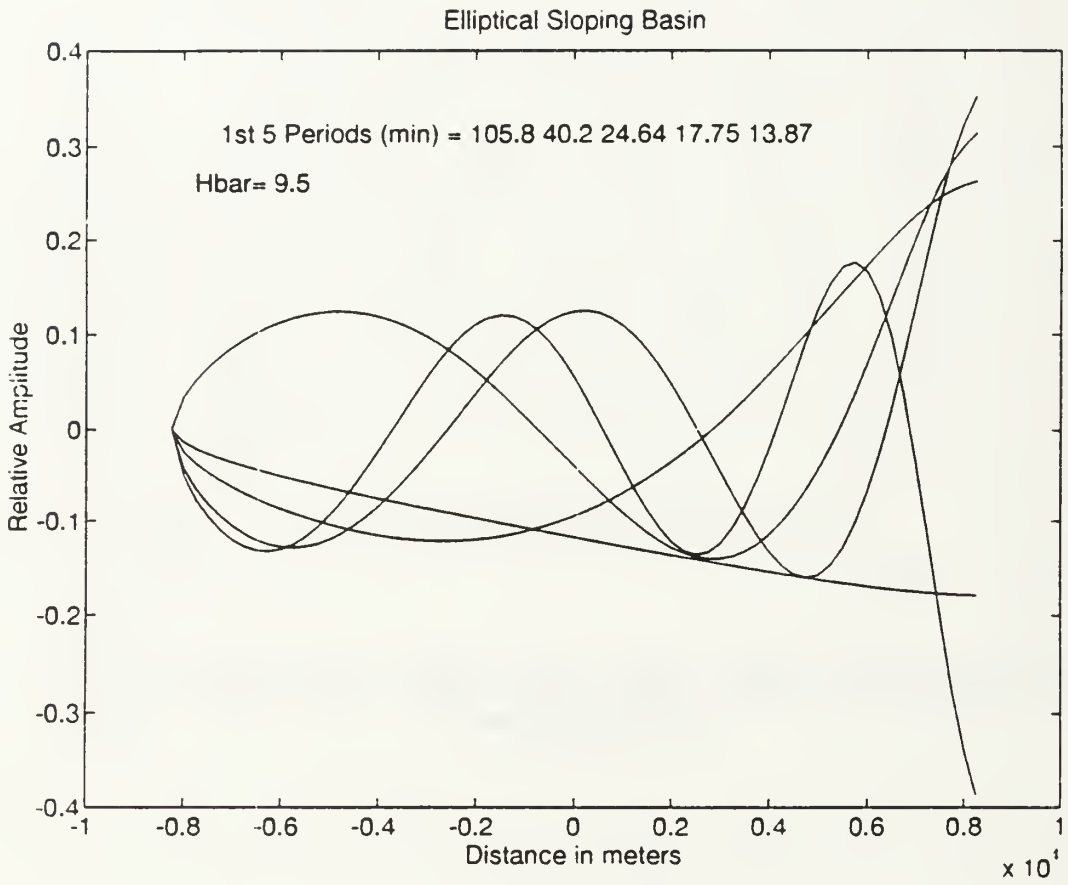


FIGURE 11. Numerical solution with an elliptical boundary and a uniformly sloping bottom. Model results are far different from what should be the true seiche modes.

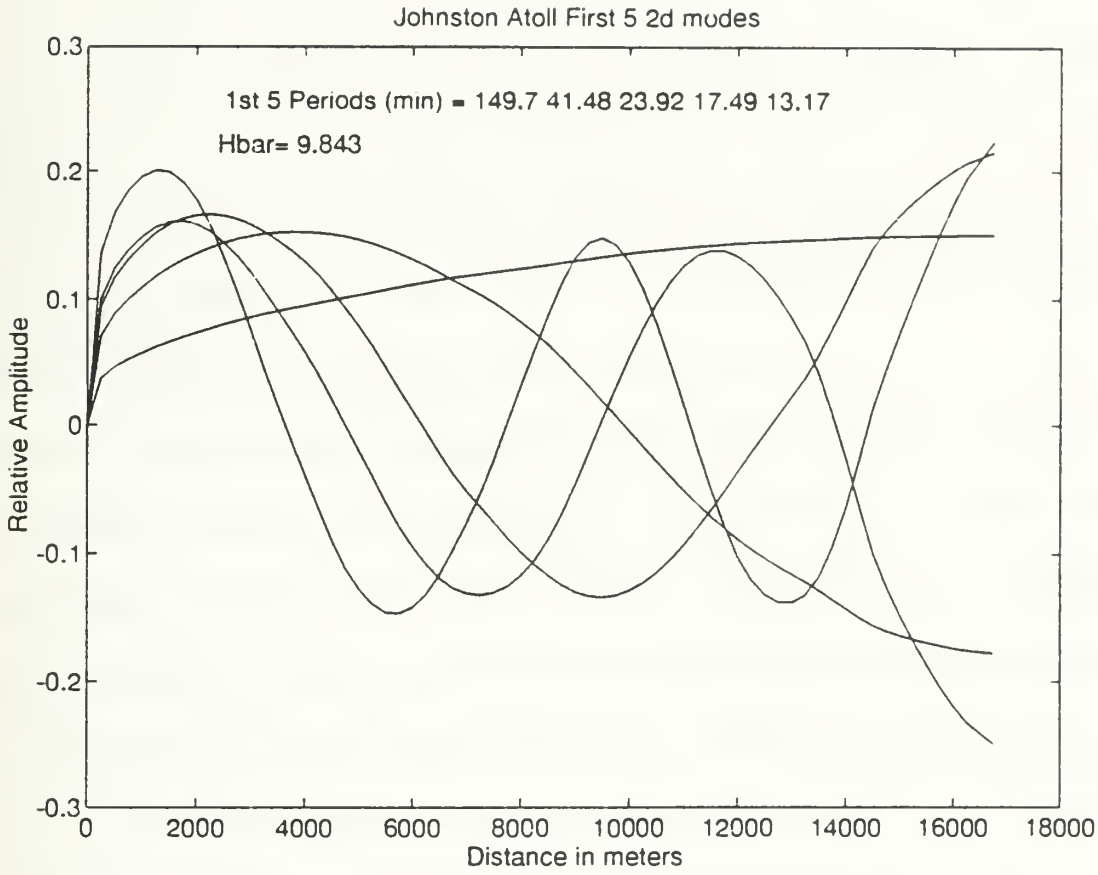


FIGURE 12. Numerical solution from the Johnston bathymetry. Note the differences between the true bathymetry and the other model approximations.

approximations are made to the lagoon. In the true bathymetry case, modal shapes are distorted from purely sinusoidal shape of the rectangular flat bottom case implicit in the analytic solution.

B. THREE-DIMENSIONAL MODEL RESULTS

Extensive trouble shooting, debugging, and evaluation of model runs that varied in resolution, application of boundary conditions, and geometric approximations of the lagoon were conducted. As discussed earlier, the stability of the solution is related to the eigenvalue solver's ability to solve large matrices. Advanced numerical techniques to condition large non-symmetric eigenvalue matrices would improve the numerical stability, resulting in the ability to increase the resolution of the grid. Saab (1989) has developed these techniques, however, they currently do not reside within the numerical MATLAB (4.0) package. A resolution of 21 X 13 results in a matrix of about 240 X 240. This compromise resolution seemed to deliver the best results for this application. The input matrices for this resolution are included in the program diskett attached to the Appendix. Parameters generated from that data set result in a maximum value of μ of .733 and a center to focus distance of approximately 6300 meters. Proper application of the depth dependent boundary condition is of critical importance. Variations in depth along the boundary can result in significant differences in numerical computations if the boundary conditions are not properly applied. For this reason,

a discussion of the techniques devised to determine the proper boundary values is included in the documentation section of the Appendix.

Figures 13 - 20 show geographical representations of model generated free surface elevation of the most energetic modes. Tables 3 and 4 list the numerical results obtained through the execution of the three-dimensional seiche models. If these shapes are characteristic, reasonable expectations of the oscillating free surface shape, then the location of nodes (lines of zero elevation) and anti-nodes (elevation maximums and minimums) generated by the model are useful in examining current flow, pressure variations and in the placement of instrumentation for future deployments.

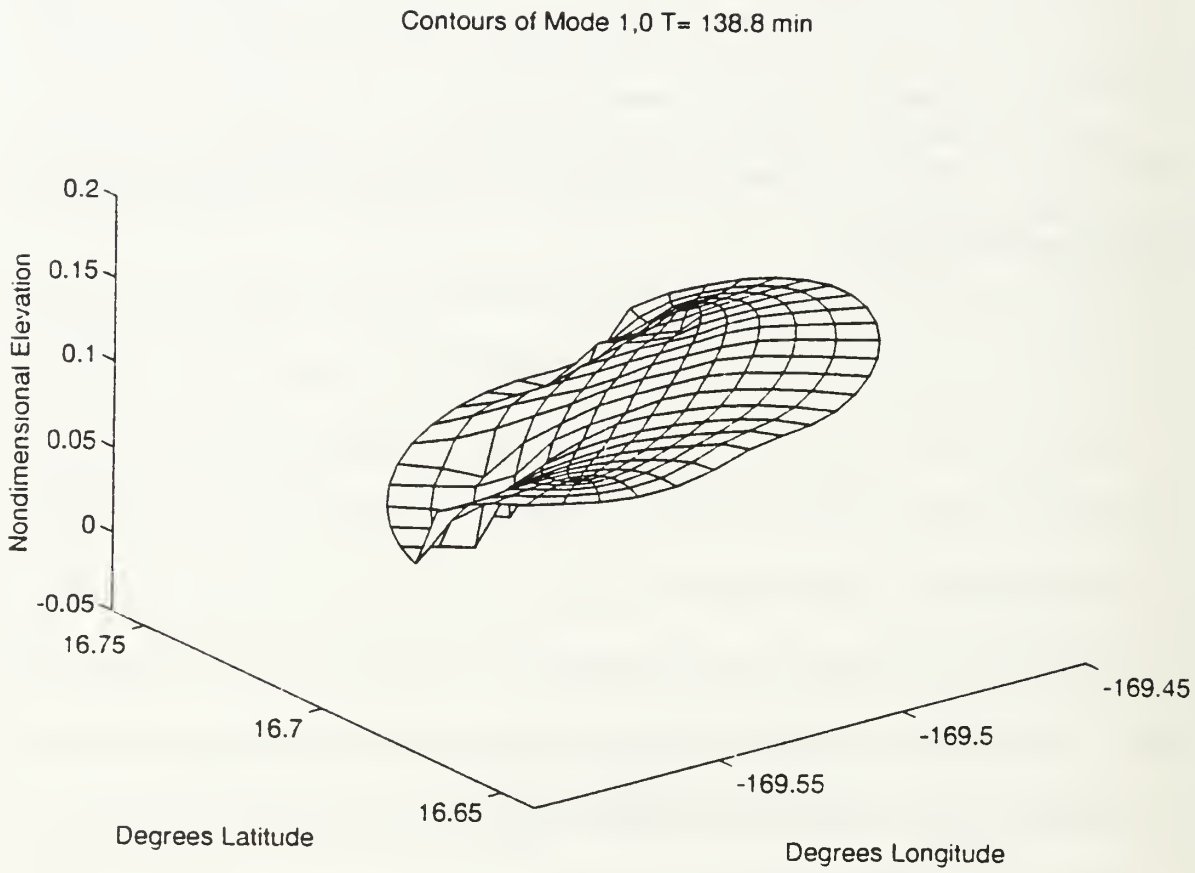


FIGURE 13. Three-dimensional rendition of the model generated sea surface elevation for mode (1,0). Elevation is nondimensional.

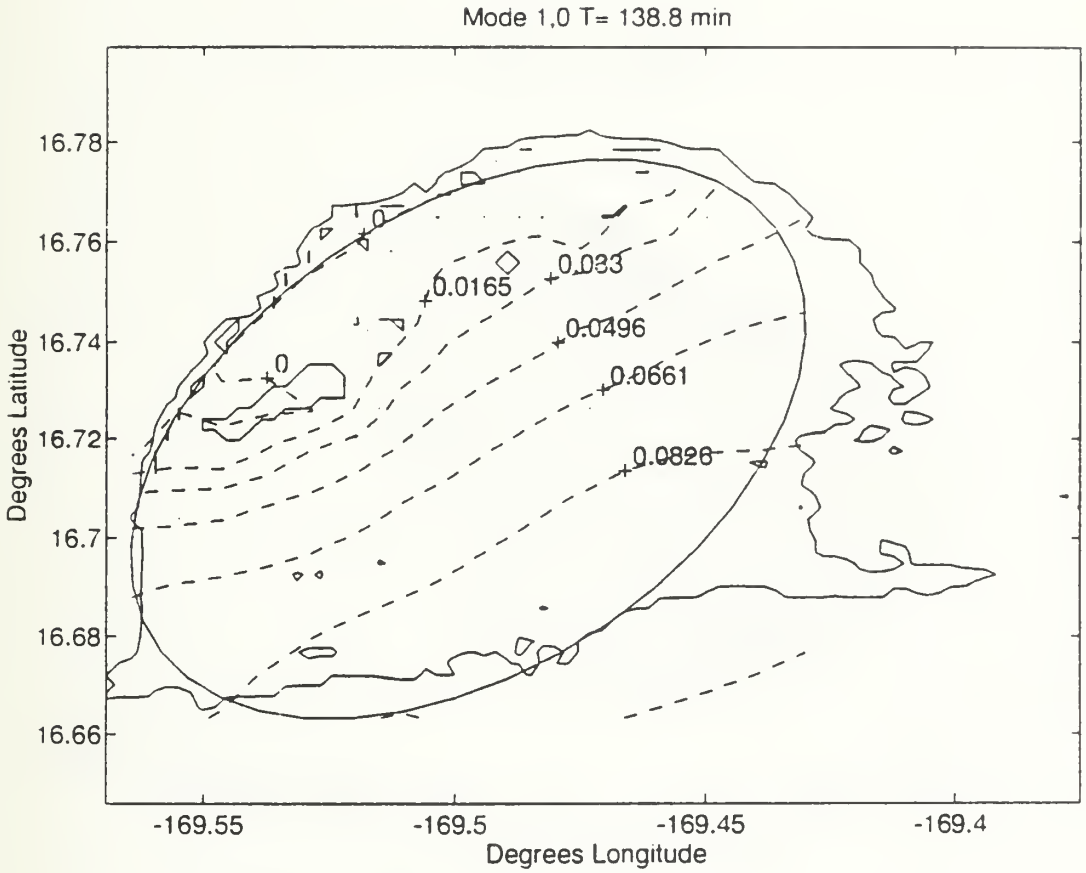


FIGURE 14. Contours of Mode (1,0) from the three dimensional elliptical model overlaid on the bathymetry. The node line is labeled 0 and crosses the lagoon between the Island and the reef. Mode(1,0) energy was observed in several data sets.

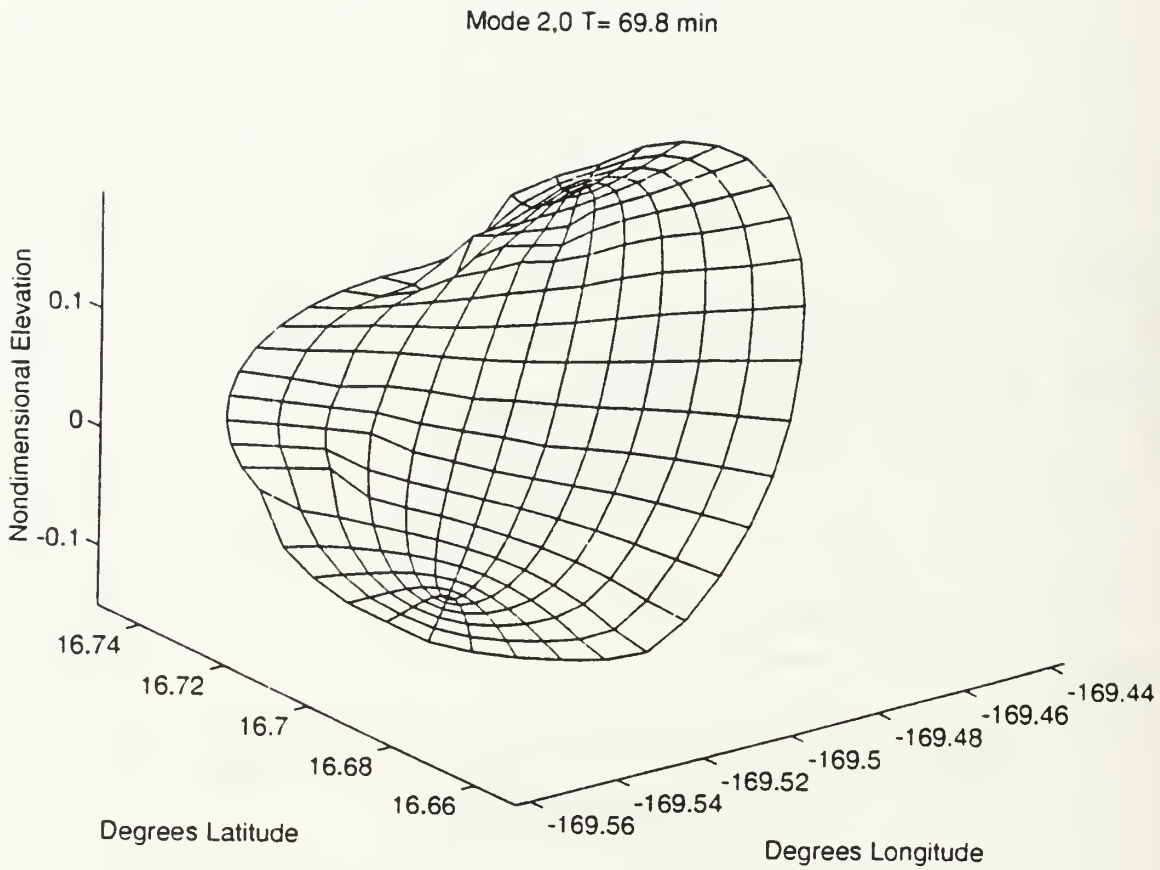


FIGURE 15. Three-dimensional rendition of the model generated sea surface elevation for mode (2,0). Elevation is nondimensional.

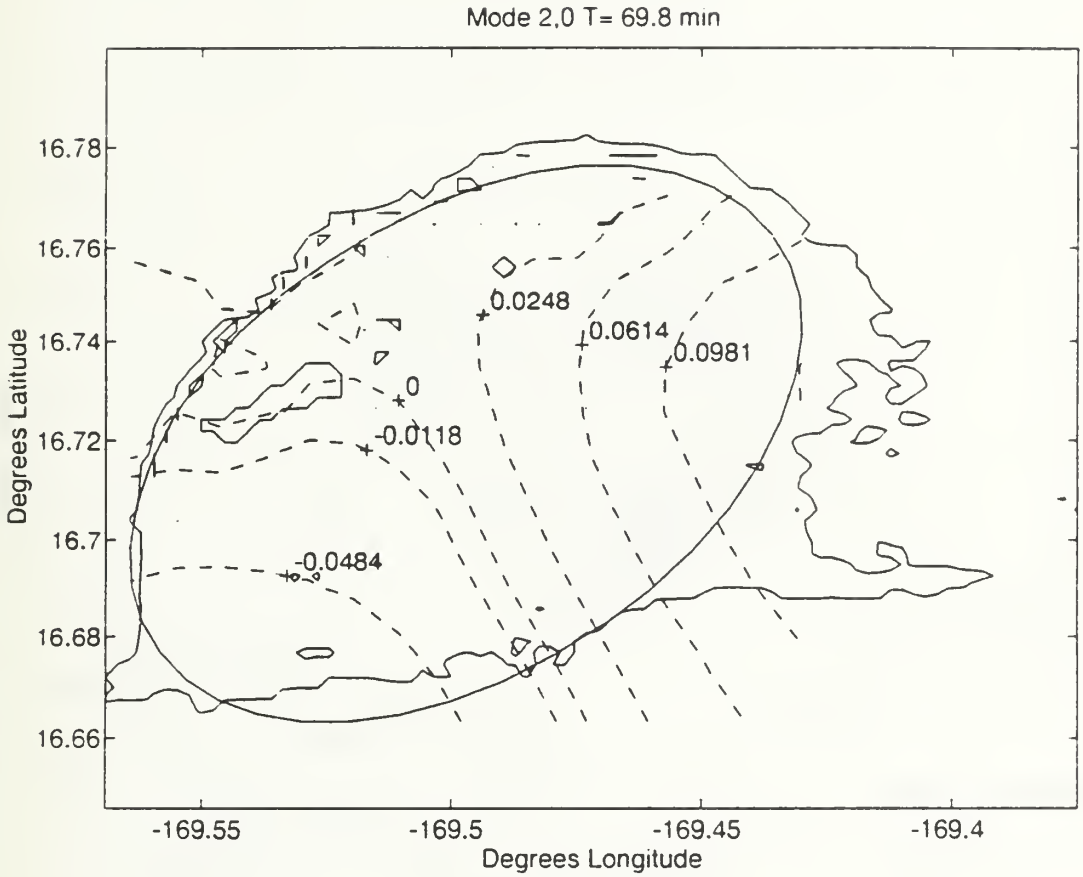


FIGURE 16. Contours of Mode (2,0) from the elliptical model. The node line again crosses the lagoon between the Island and the reef. Some evidence exists that mode (2,0) energy may be observable.

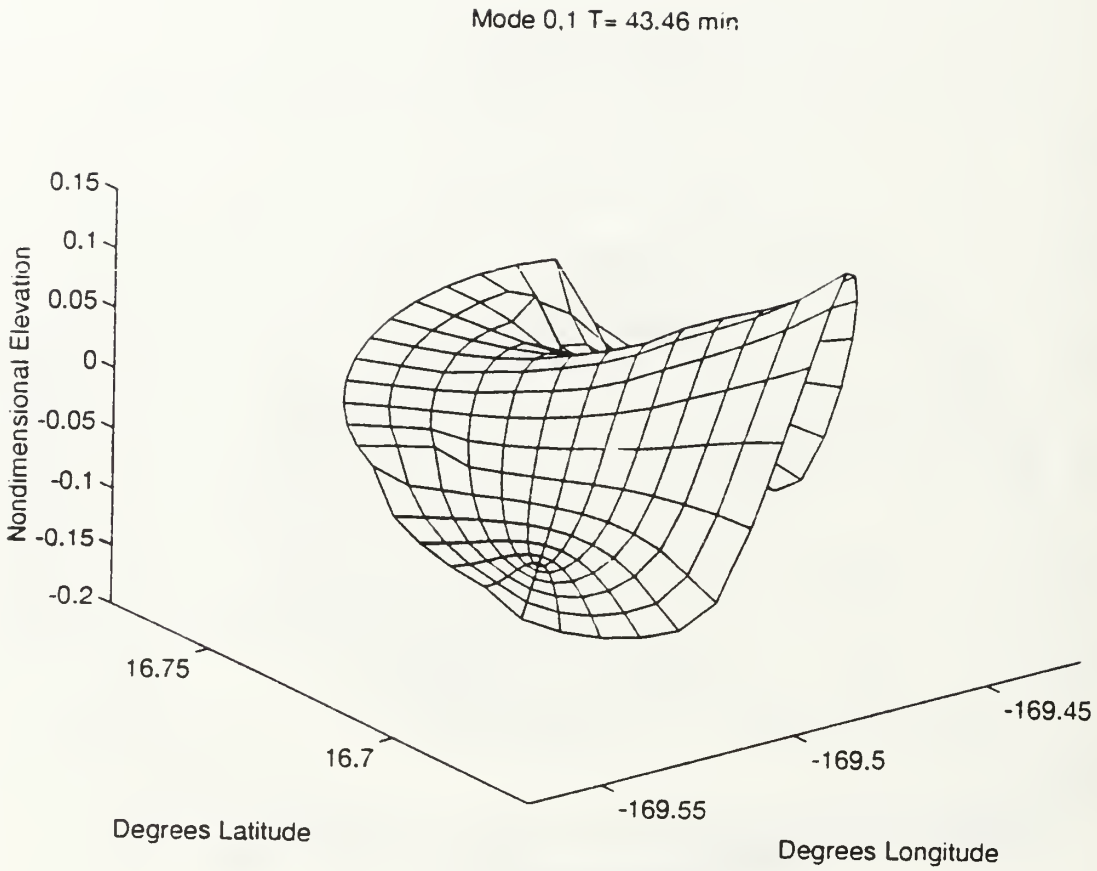


FIGURE 17. Three-dimensional rendition of the model generated sea surface elevation for mode (0,1). Elevation is nondimensional.

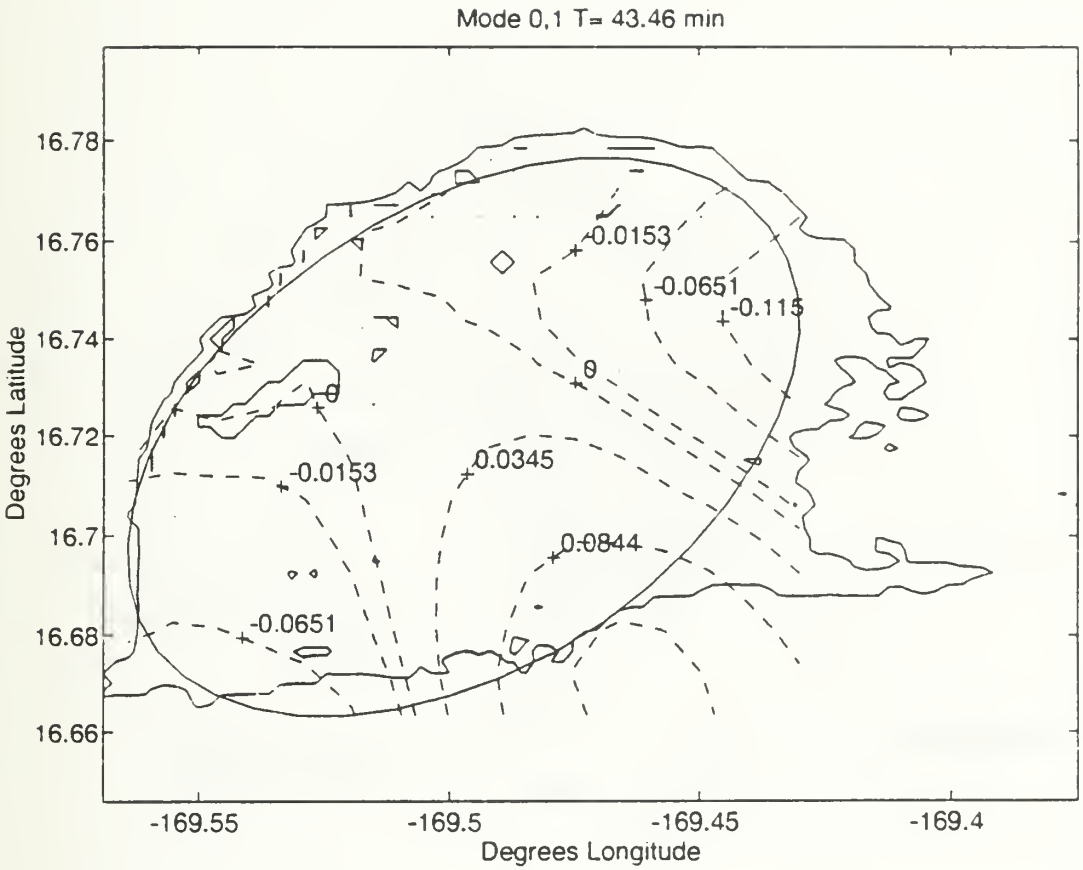


FIGURE 18. Contours of mode (0,1) numerical solution overlaid on the bathymetry.

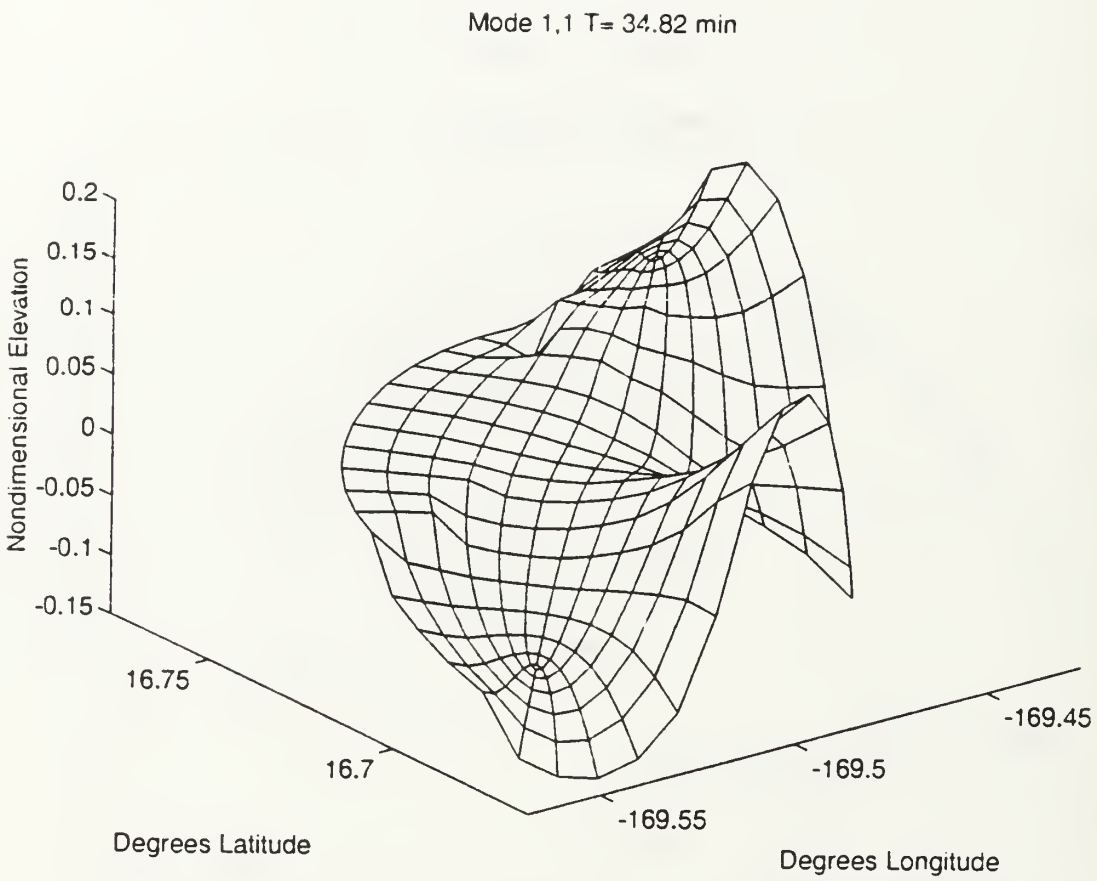


FIGURE 19. Three-dimensional rendition of the model generated sea surface elevation for mode (1,1). Elevation is nondimensional.

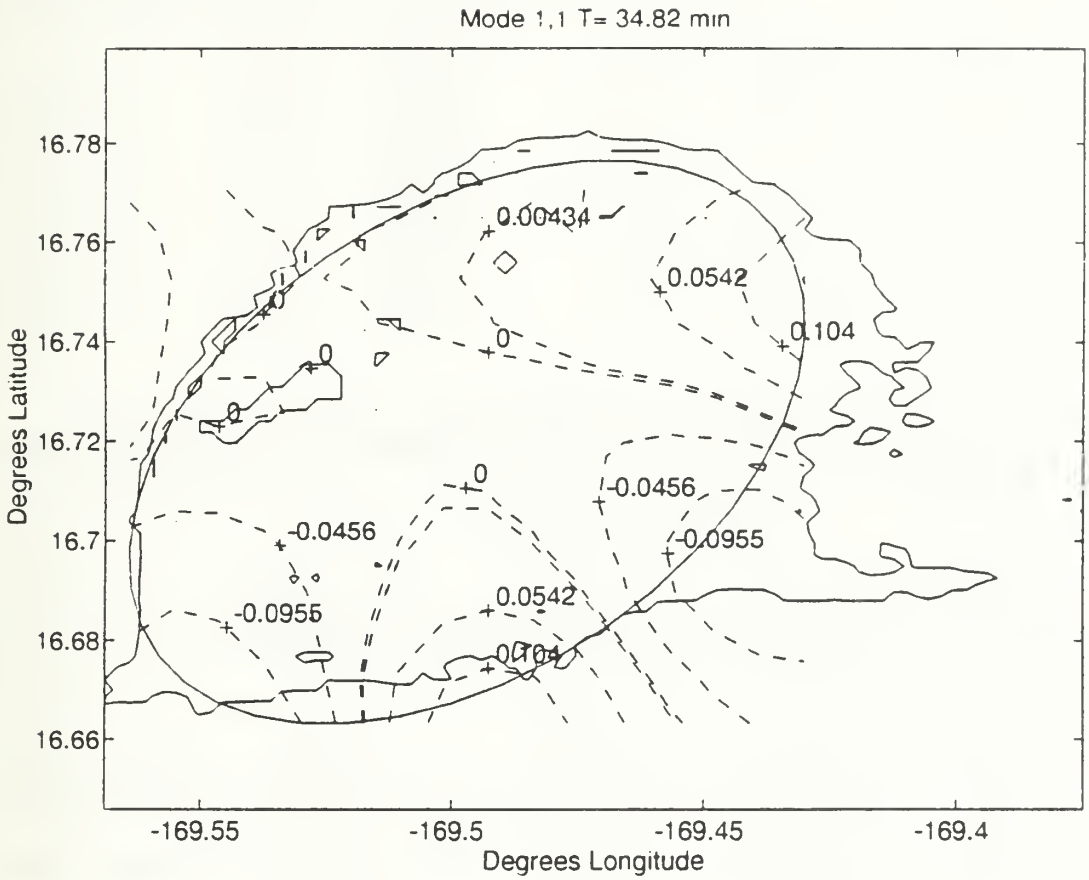


FIGURE 20. Contours of mode (1,1) numerical solution overlaid on the bathymetry.

TABLE 2. MODEL RESULTS OF TWO-DIMENSIONAL GEOMETRIC APPROXIMATIONS.

Periods in minutes

Modes	Analytic uniform	Rect. uniform	Rect. sloping	Ellip. uniform	Ellip. sloping	Ellipt. Bathymetry
0	115.10	115.21	103.90	125.30	105.80	149.75
1	57.55	38.41	44.94	38.10	40.22	41.49
2	38.37	23.05	25.05	22.61	24.69	23.90
3	28.78	16.47	18.09	16.19	17.75	17.50
4	23.02	12.83	14.06	12.60	13.87	13.17
5	19.18	10.28	12.26	10.24	11.73	10.84
6	16.44	8.69	10.39	8.67	9.46	9.18
7	14.39	7.54	9.02	7.53	8.36	8.09
8	12.79	6.67	7.97	6.65	7.58	7.18
9	11.51	5.97	7.15	5.96	6.58	6.41

TABLE 3. MODEL RESULTS OF THREE-DIMENSIONAL RECTANGULAR APPROXIMATIONS

Periods in minutes

Mode	Analytic uniform	Rectangular uniform	Rectangular sloping
1,0	115.78	114.45	147.01
2,0	57.89	82.99	98.40
0,1	56.14	65.59	72.11
1,1	50.52	55.80	59.30
2,1	40.30	53.04	52.23
3,0	38.60	49.99	47.89
3,1	31.80	46.29	47.23
4,0	28.95	43.79	45.00
4,1	25.75	43.25	43.01
5,0	23.15	42.03	41.58

TABLE 4. MODEL RESULTS OF THREE DIMENSIONAL ELLIPTICAL GEOMETRIC APPROXIMATIONS

Modal periods in minutes

Mode	Analytic Uniform	uniform Flat	True Bathymetry
1,0	115.89	116.15	138.86
2,0	57.89	70.12	69.81
0,1	56.14	53.56	43.56
1,1	50.52	45.49	34.82
2,1	40.30	41.46	31.86
3,0	38.59	40.94	30.37
3,1	31.80	38.07	27.19
4,0	28.94	30.87	25.26
4,1	25.73	26.19	25.04
5,0	23.16	23.10	24.46

Now that we have the fundamental periods and modal shapes of the seiche numerically estimated to some accuracy, existing historical current meter data can be analyzed to determined what frequencies play important roles in the lagoon circulation, and establish whether resonant forcing of the seiche is a significant part of the circulation. The next chapter presents a preliminary analysis of the existing data to examine the energy content and frequency distribution of current meter observations. From this information, some generalities of the observed currents are made.

IV. ANALYSIS OF CURRENTS

Now having reasonable estimates of the theoretical seiche, examination of amplitude and frequency information in current meter records should confirm the major contributing energy components. If the spectral energy of velocity and pressure reveal oscillations of statistically significant magnitude closely matching the numerically calculated seiche periods, then we will have succeeded in identifying the major sources of energy. We will also have validated the seiche model and determined which, if any, of the theoretical components are important to the circulation within the lagoon.

A. DATA AND METHODS

1. Current Meter Data Collection

Current meter records became available in February 1994. Moorings had been placed primarily in the lagoon between the northwest edge of the main island and southeast of the reef. Of the 24 records obtained, eighteen were of sufficient length and adequate sampling to be useful. Only four were of greater than 4 days duration and only five records had reliable pressure information. Of the 18 records, thirteen were sampled at 15 minute intervals and five were sampled at 30 minute intervals. Figure 21, displays current meter locations. Table 5 lists current meter position and deployment dates.

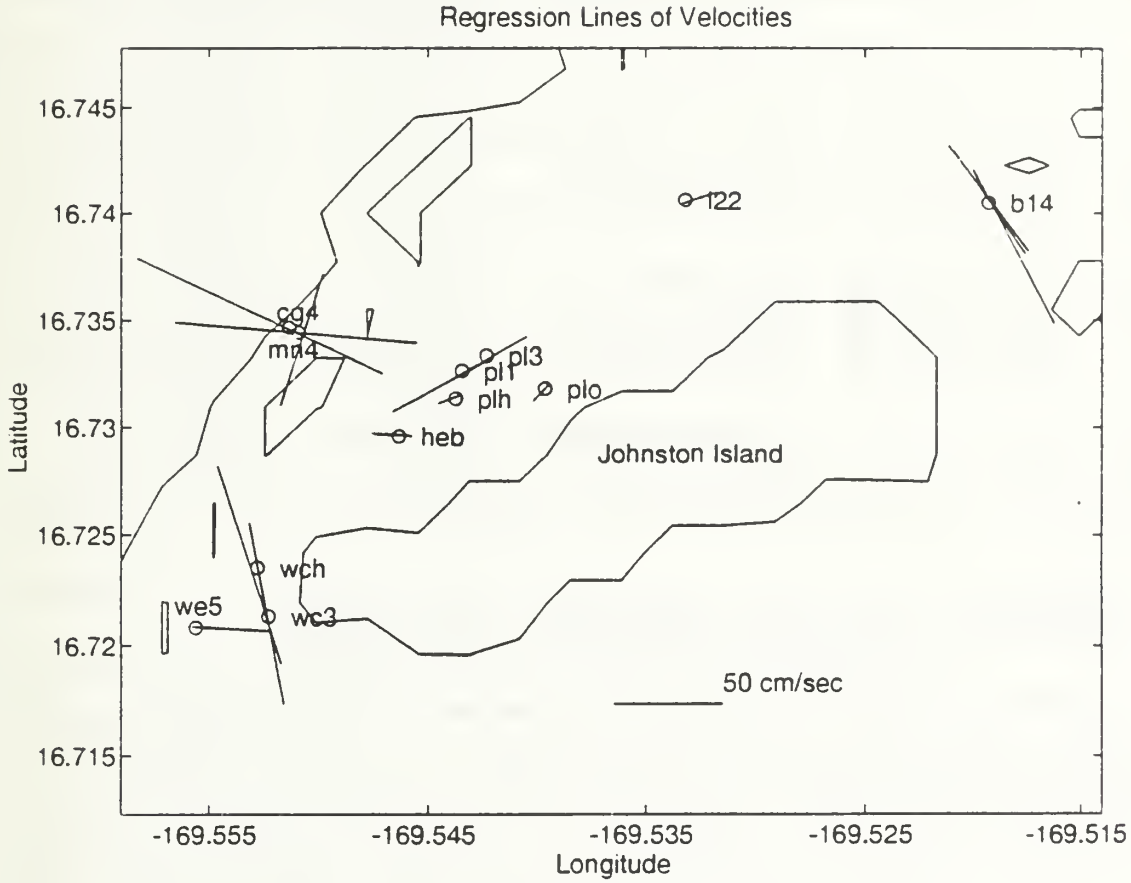


FIGURE 21. Current meter locations are depicted as circles. Lines represent the axis of the flow as computed in a least squares regression. Flow at all locations was highly polarized. More than one line indicates multiple observations at the same location. The flow pattern is distinct. Current flow around the island is leaked through Monson's gap, MN4, and channelized through WCH and B14.

TABLE 5. CURRENT METER LOCATIONS.

Buoy	Geographic Location		Date Deployed	Date Recovered	sampling Rate
L22	16° 44.61 N	169° 31.96 W	12/12/91	2/22/91	30
B14	16° 44.43 N	169° 32.78 W	12/12/91	12/22/91	30
			7/23/93	7/27/93	15
			10/28/93	11/5/93	15
PL0	16° 43.91 N	169° 33.02 W	7/4/93	7/7/93	15
			7/10/93	7/18/93	30
PL1	16° 43.96 N	169° 32.61 W	7/4/93	7/7/93	15
WCH	16° 43.63 N	169° 33.02 W	7/4/93	7/7/93	15
HEB	16° 43.76 N	169° 32.78 W	7/3/93	7/7/93	15
			7/10/93	7/13/93	30
PL3	16° 44.00 N	169° 32.54 W	7/23/93	7/27/93	30
PLH	16° 43.88 N	169° 32.62 W	7/12/93	7/18/93	15
WC3	16° 43.59 N	169° 33.03 W	7/23/93	7/27/93	30
MN4	16° 44.08 N	169° 33.08 W	8/2/93	8/6/93	15
			10/13/93	10/15/93	15
CG4	16° 44.07 N	169° 33.06 W	8/2/93	8/6/93	15
WE5	16° 43.51 N	169° 33.21 W	10/13/93	10/15/93	15

B. SPECTRAL ANALYSIS - FREQUENCY DEPENDENCE

Each time series of current speed and pressure from the eighteen records having sample length greater than 256 observations was spectrum analyzed in the usual manner. The data was detrended and a modified Hanning window was applied. Due to the short record lengths, a direct Fourier transformation was performed and then, energy density spectra computed. The individual energy spectra of current magnitudes were then frequency averaged. Frequency averaging of the ensemble gives a basin typical indication of current energy. From this average spectrum, modes of the peaks in the spectrum determined the relative

contribution of each component. The frequency averaged energy spectrum of current magnitude is shown in Figure 22, and the frequency averaged energy spectrum of pressure is shown in Figure 23.

As accurate tide gauge measurements were not available, tidal amplitude estimates were obtained through commercially available computer software, Micronautics, Inc (1994), using a time delay offset from the 1994 database in Hawaii. Mean amplitudes are approximately one meter, referenced to MSL. The frequency spectra of a three month time series of model generated tidal amplitude is shown in Figure 24. Components of the model generated tide closely coincide with the observations and serve as a baseline spectrum for comparison.

The two main components readily identified are the principal lunar-solar diurnal (K1) and the principal lunar semi-diurnal (M2). These 2 components account for the bulk of the energy within the bay. At the low end of the spectrum, the energy peak associated with the inertial period is prominent.

The peaks not specified by predicted tidal, inertial, or seiche are assumed to be either modulation (sums or differences of components) or harmonics (multiples of a single component). Calculations show that the first harmonic of the semi-diurnal (M2) tide (noted as 2*M2), accounts for the intermediate energy peak with a center frequency of about 1/6 cph. The peak with a center frequency at approximately 1/4 cph is the second harmonic of the semi-diurnal (M2) tidal frequency (3* x M2). The third harmonic of the M2 tide is also evident at about 1/3

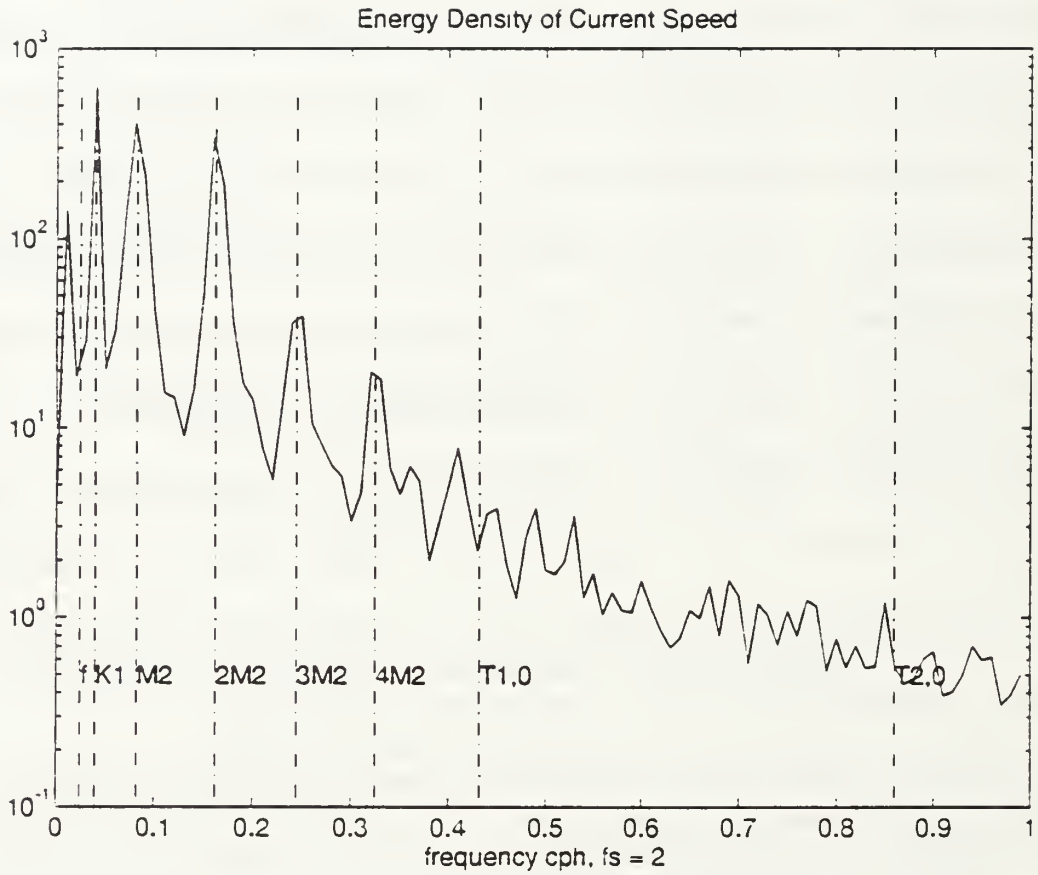


FIGURE 22. Spectral energy of current magnitude (cm/s)^2 . The model generated frequencies of tidal, inertial, and seiche energy are depicted as vertical lines.

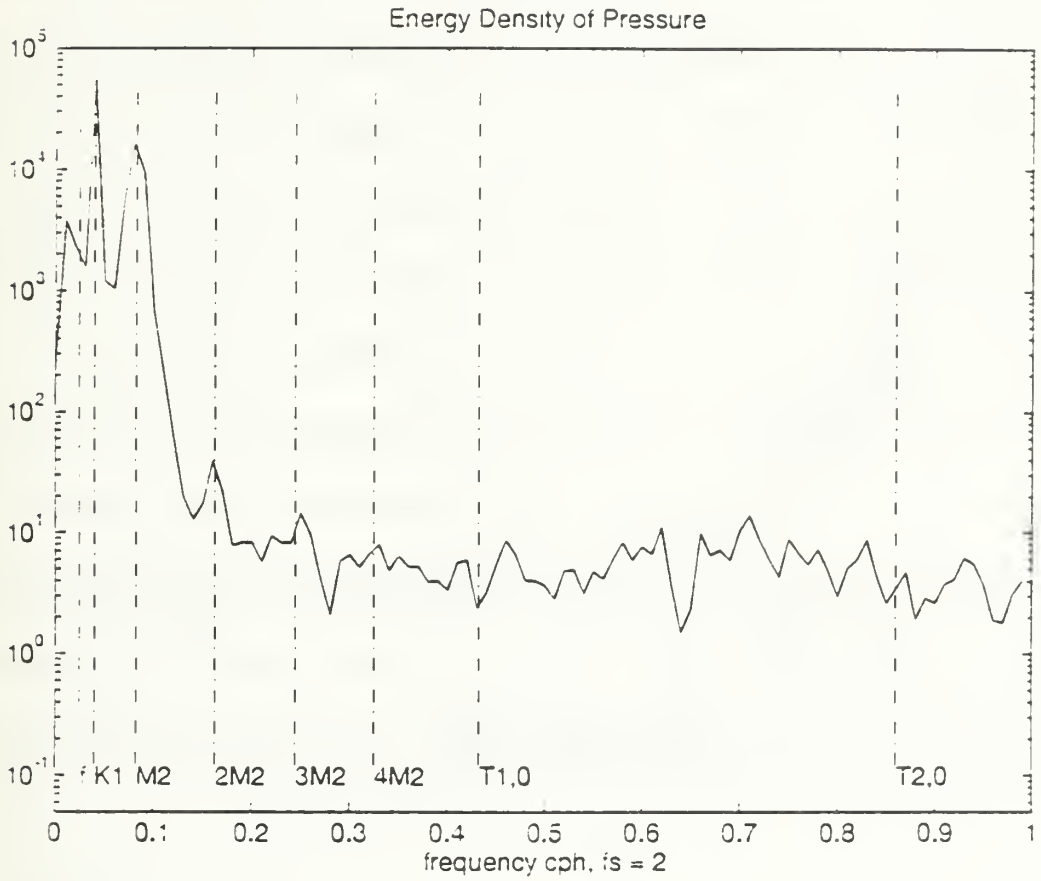


FIGURE 23. Ensemble average energy density of pressure within the lagoon. No significant observations of seiche was found in the pressure records.

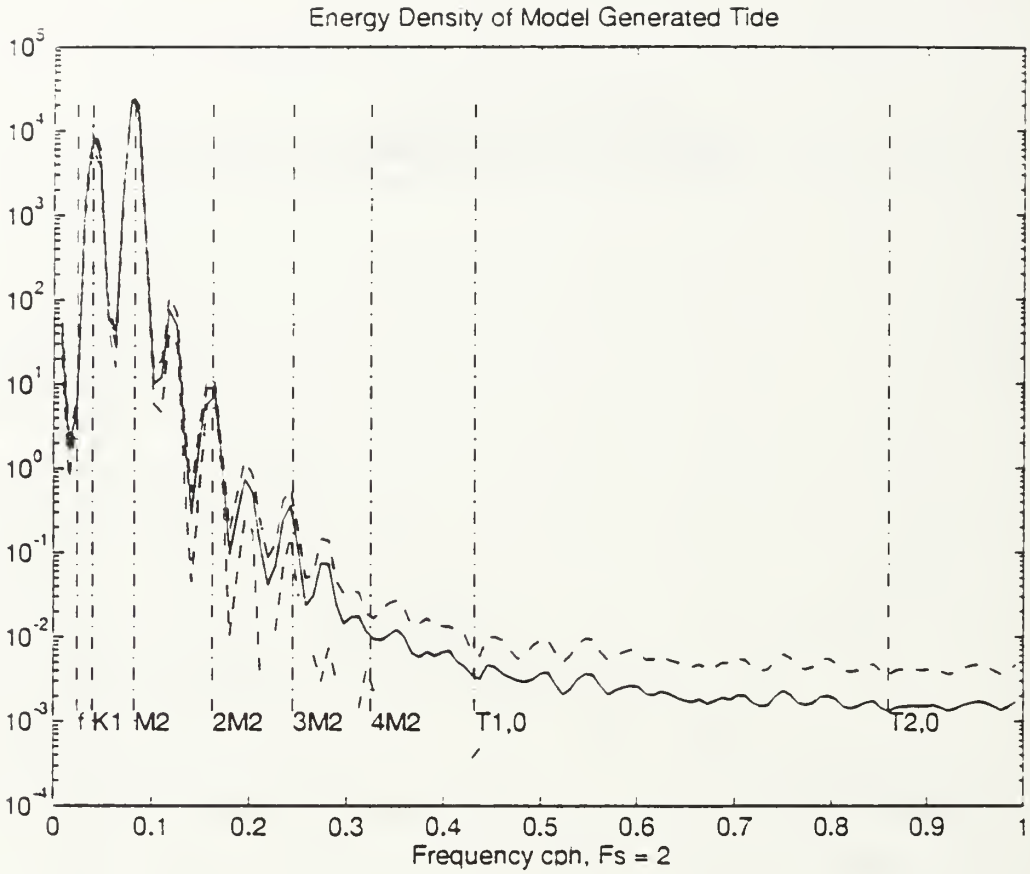


FIGURE 24. Spectral energy of a three month time series of model generated tidal amplitude. This serves as a baseline for computations of spectral energy in the observations of current.

cph. The fundamental seiche mode appears in several records, the most evident being station I22, Figure 25. This record was the longest of the current meter records and contained what appears to be the most complete indications of the bay's natural oscillation. The energy peak has a center period of 140 minutes, very close to the calculated fundamental seiche period identified as T(1,0). Station B14 during the July 1993 deployment showed strong seiche mode energy in the V component of velocity, centered again near the 140 min period T(1,0). At other locations and times, seiche mode energy was not nearly as significant.

It appears that very little energy exists below the fundamental seiche mode. For those records sampled at 15 minute intervals, 95 percent confidence interval calculations using a X^2 distribution, showed that high frequency energy (above one cycle per hour) was not significantly different from zero. Some evidence exists, however, to conclude that the second seiche mode of about 70 minutes contributes some energy. The basis for this conclusion is in the consistency with which this peak, though small in magnitude, arises in the energy density spectra of individual records.

No significant seiche mode energy was found in any of the pressure records. As wind data were not available, wind induced currents were impossible to define. However, it is surmised that wind forcing would result in energy peaks spanning or coincident with longer periods than the semi-diurnal tide and could excite the seiche.

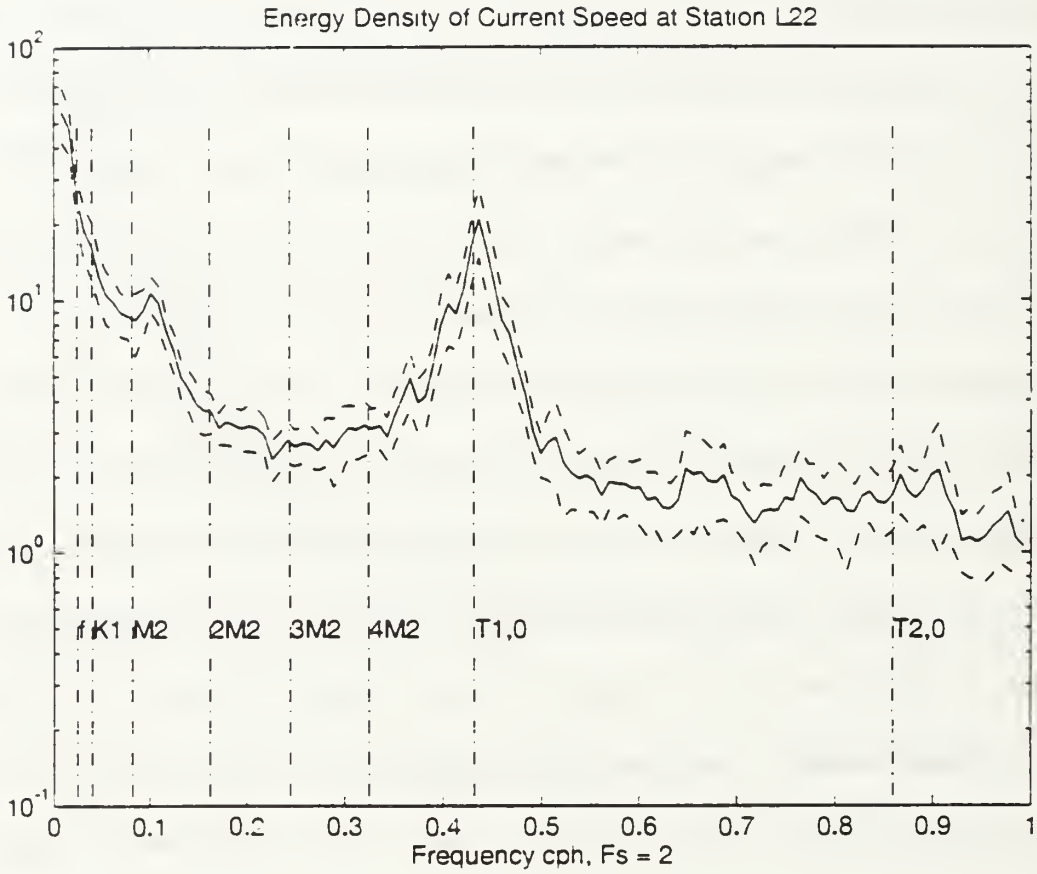


FIGURE 25. Manifestation of seiche in the energy density of currents at Station L22. Modal generated seiche periods are depicted as lines.

Table 6 summarizes the periods of oscillation identified in this study as important to the physical oceanography of the Johnston Atoll lagoon. They should serve as a basis for future modeling research into the Island's circulation.

TABLE 6. MAJOR COMPONENTS OF ENERGY.

Component	period hours	frequency cph	Relative Amplitude
f	41.57	.0241	27
K1	25.36	.0394	100
M2	12.4	.0820	67
2*M2	6.16	.1623	55
3*M2	4.09	.2443	6.3
4*f	3.08	.3247	3.1
T1,0	2.31	.4323	1.3
T2,0	1.16	.8594	0.2

C. CURRENTS IN GENERAL

The observed flow at all locations except L22 is highly polarized and phase locked to the tidal cycle. Cross spectral analysis between the pressure fields and the components of velocity indicate a phase lag of about 160 degrees throughout the lagoon. The U component of velocity is uniformly westward during the rising tide and uniformly eastward on the ebb. V components are uniformly northward during tidal rise and southward on the ebb. Figure 26, depicts the U and V components of velocity, superimposed on the recorded pressure amplitude at station MN4. Phase offset varies only slightly from station to station, consistently maintaining an approximate 160 degree lagging phase lock in both U and V.

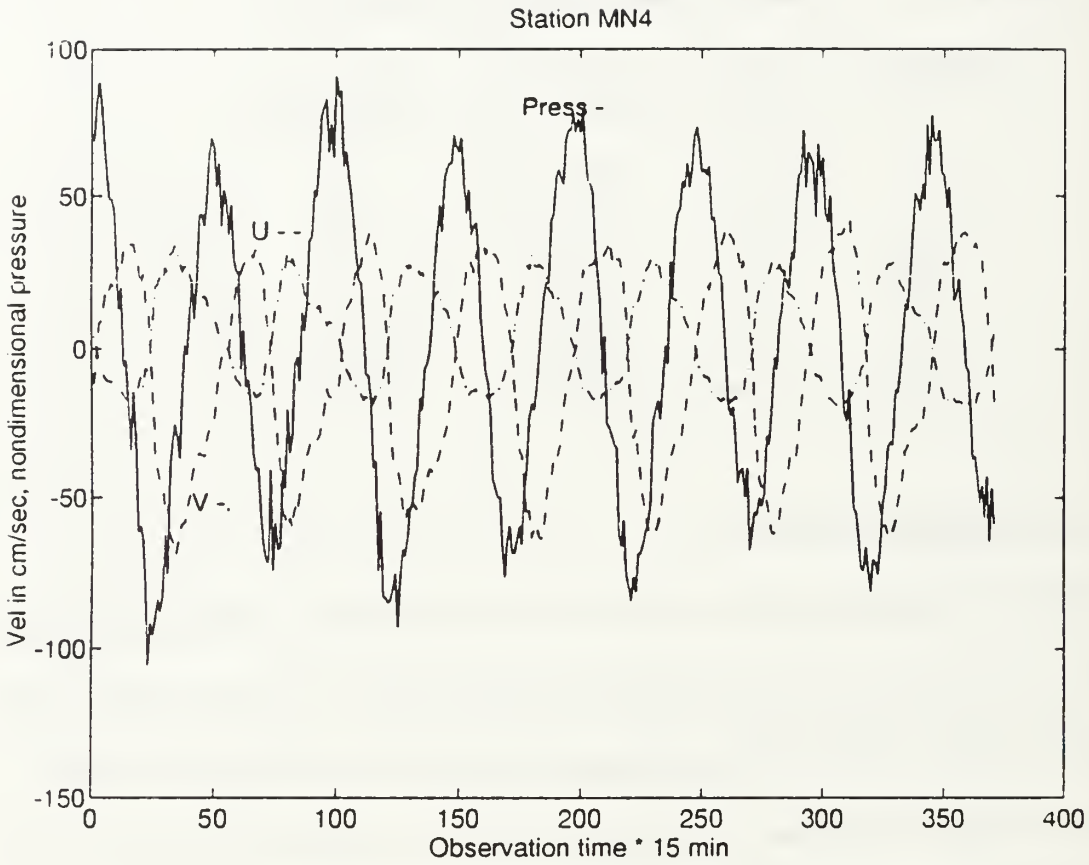


FIGURE 26. U, V, and pressure at station MN4 showing phase offset of approximately 160 degrees. Typical of the entire basin, U and V components are West and North following the tidal rise and South and East during the ebb. Tidal amplitude is in cm from msl.

The polarity of the current axes strongly indicates the importance of tidal forcing within the confines of the lagoon and at the entrances through the reef. Figure 27 depicts compass representations of velocity vectors at selected locations. Mean current magnitudes at each location range from less than 2 cm/sec directly in the lee of the island to more than 30 cm/sec in the channels. The largest currents measured were at B14 and at Monson's Gap where the peak speeds exceeded 60 cm/sec. Because the flow indicates strong polarity, a least squares linear regression gives the most accurate estimate for averaging flow direction and magnitude. Figure 21 depicts regression lines scaled to 50 cm/sec and superimposed on the current meter locations.

Assuming temporal and spatial stationarity of the data sets, it is possible to qualitatively estimate general current patterns. On the incoming tide, currents are generally cyclonic (counter clockwise) around the island. Following the onset of the tidal ebb, currents reverse, becoming anticyclonic (clockwise). Significant exchange between the interior lagoon and the open sea takes place at Monson's Gap. Stick plots of the time series of current are shown in Figure 28 through Figure 33. Velocity vectors are plotted as a function of observation (n x dt).

There appear to be wind event scale current motions in the observations that coincide with the most dramatic manifestations of seiche. Of note is the large scale wind event pushing a southward flow at station B14 during the October to

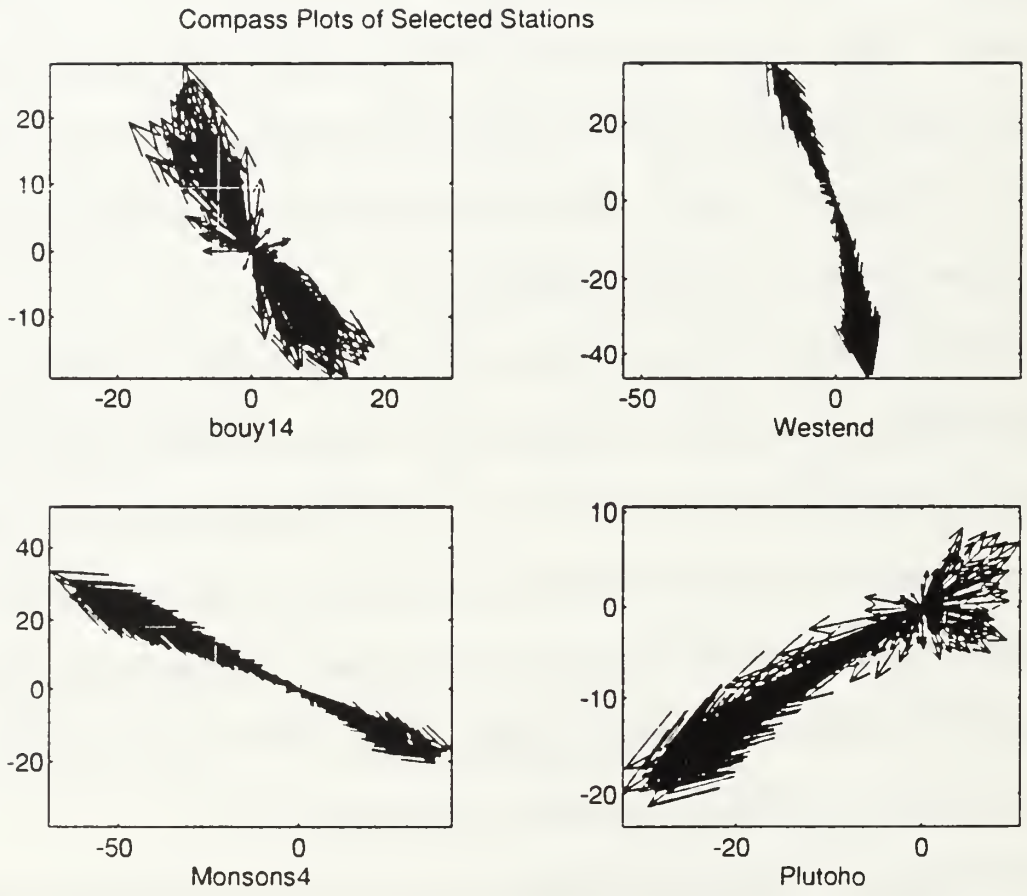


FIGURE 27 Compass representation of U and V velocity components showing the highly directional nature of the flow. Velocities are in cm/sec.

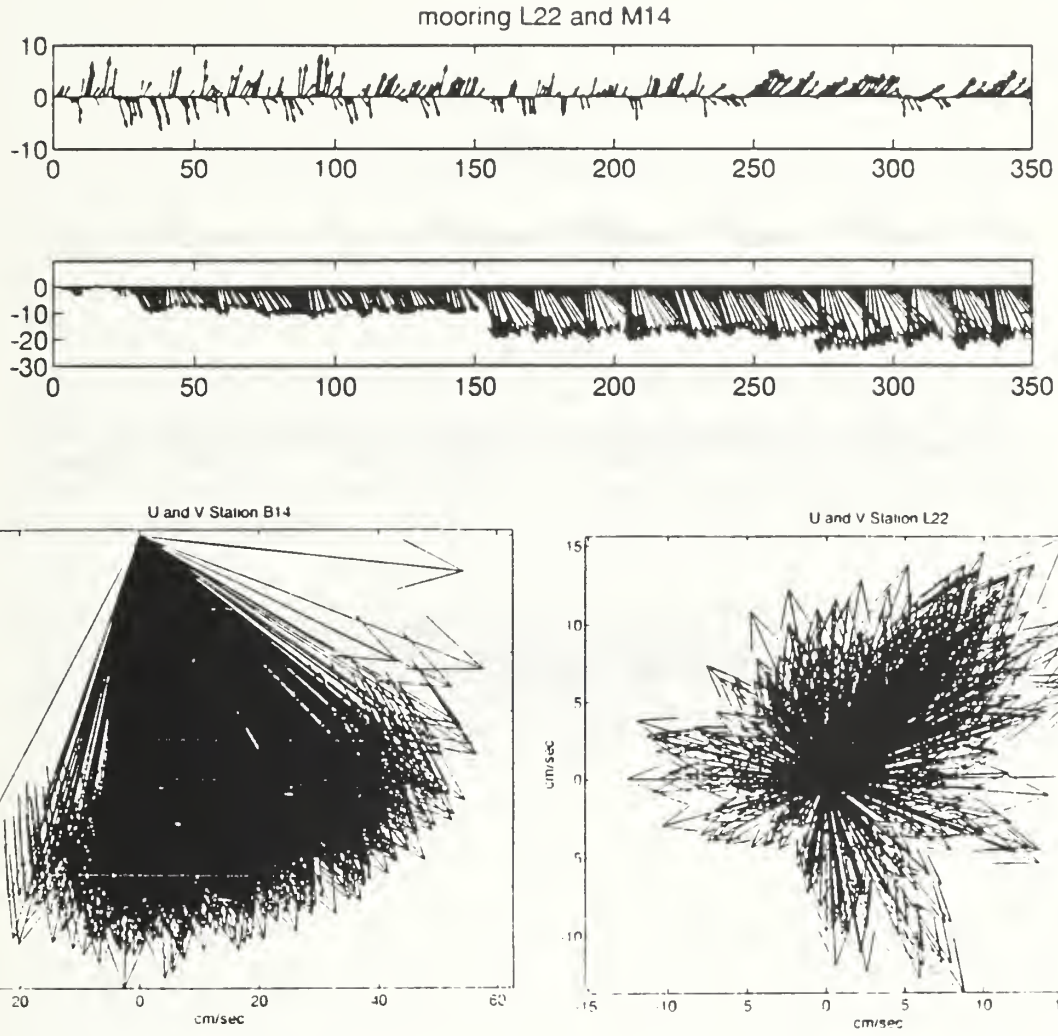


FIGURE 28. Time series and compass representation of simultaneous observations at moorings B14 and L22. Flow is southward flow at B14 and oscillates at the fundamental seiche mode at station l22.

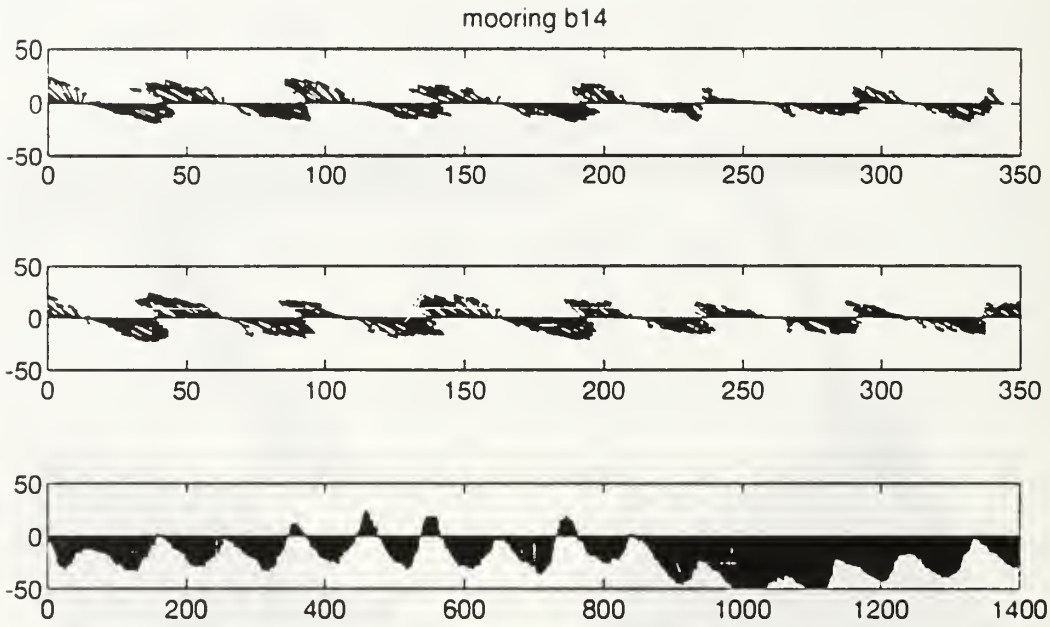


FIGURE 29. Representative time series from three observations at mooring B14 showing highly directional tidal flow. The bottom time series shows the effect in October of large scale wind event overriding the tidal flow. Each tic mark represents an observation, dt =15 min.

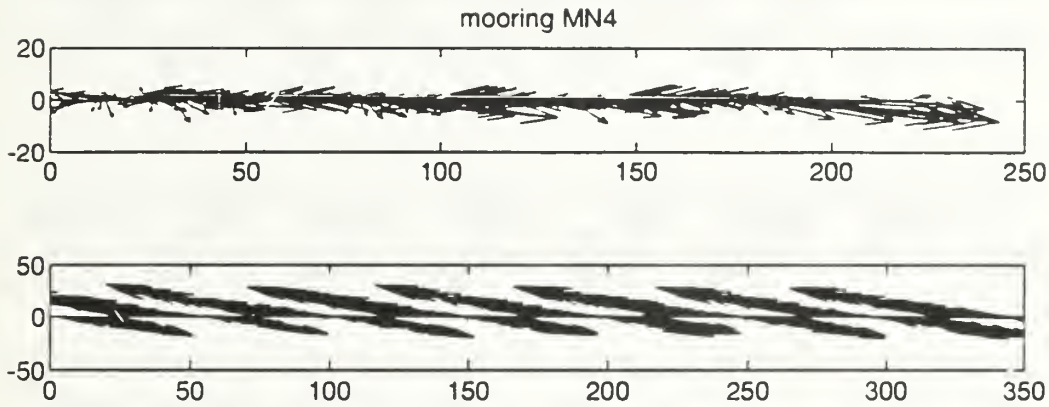


FIGURE 30. Time series of current observations for 2 different deployments at Monson's gap (MN4) where strong tidal flow enters and exits the lagoon through the reef. Each tic mark represents an observation, dt =15 min.

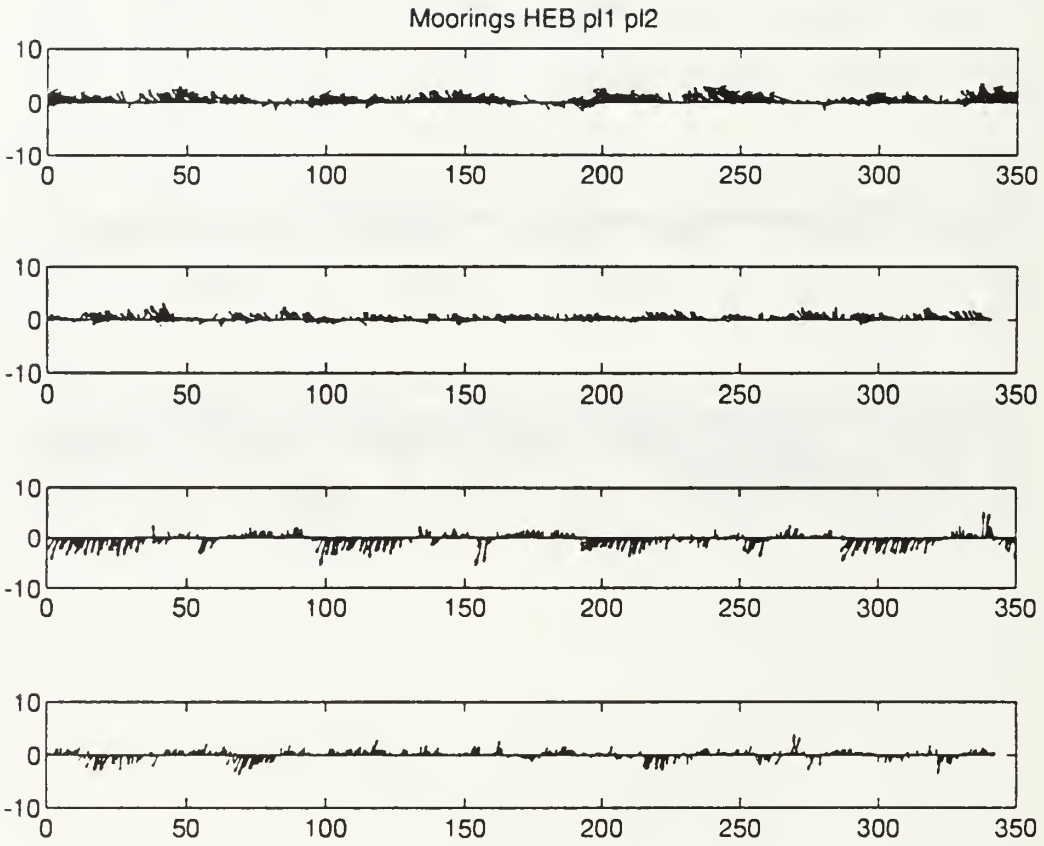


FIGURE 31. Time series of current observations. These stations are within the confines of the lagoon and reflect the quiescent nature of the protected Island lee. Each tic mark represents an observation, dt =15 min.

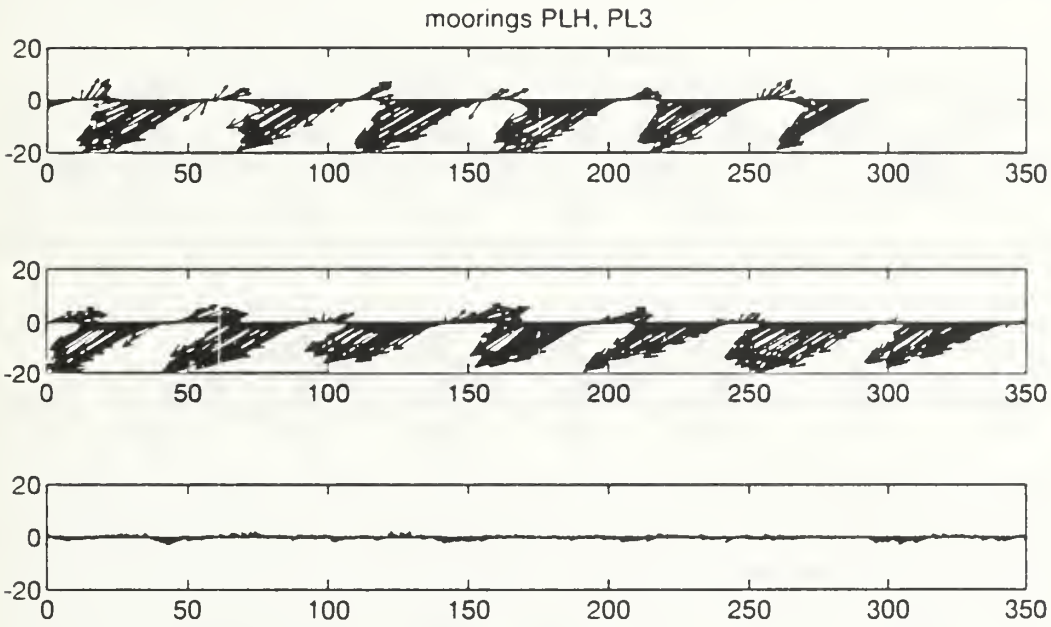


FIGURE 32. Time series of current observations. Moorings PLH and PL3 are only slightly offset from HEB and PLO, however, exhibit highly directional flow of the other more energetic locations. Each tic mark represents an observation, dt = 15 min

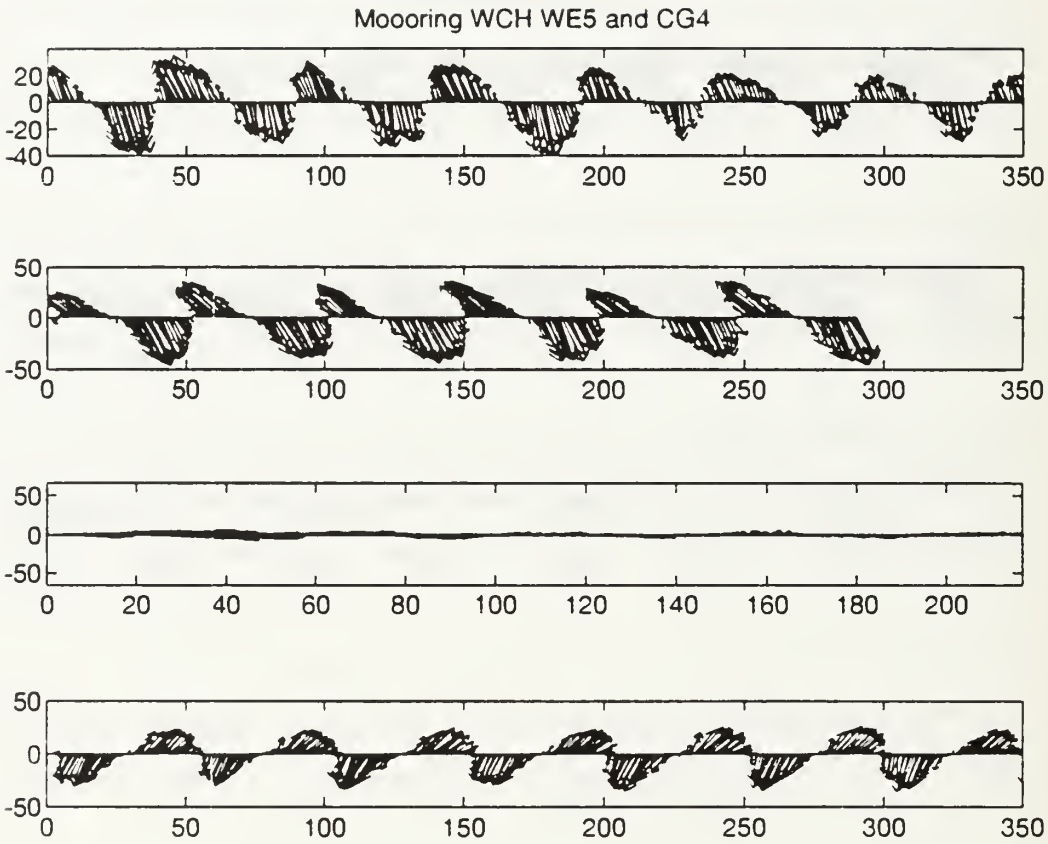


FIGURE 33. Time series of current observations. At the west end of the Lagoon, polarized flow around the island is well structured. At CG4, the flow closely parallels the East west axis of the Monsons Gap flow. dt = 15 min.

November, 1993 deployment. This flow is not tidally rectified, but consists of a mean southward flow superimposed on the highly polarized tidal flow. Station L22 was sampled during the same time as this large scale wind event and had the largest seiche mode energy of any of the data sets. This may indicate that seiche resonance may only manifest as significant in the circulation when forced by the wind.

Volumetric transport calculations for three representative station locations, B14, WCH and MN4, were computed as follows. By assuming stationarity, and that flow past the current meter is representative of the total flow across a boundary, then by choosing strategically placed current meters, an estimate of the volume transport can be made. The stations chosen represent the flow boundaries surrounding the main island. They are: station WCH, which measured flow between the main island and the reef at the west end; station MN4, which recorded flow through Monson's Gap; and station B14, which recorded flow down the main channel between the main Island and the submerged central reef. Transports, (T_s) parallel to the current flow across these lines can be calculated from the current magnitudes at these locations as follows:

$$T_s = A * \bar{s} * t$$

where A is crossectional area (distance x depth, orthogonal to current flow), $\bar{s}$ represents current speed and t is time. Let t equal to the length of one half tidal cycle, (determined from the records as times of flow reversal), s be the average speed during that period of flow direction, and A as the crossectional area between the confines of the natural barriers (i.e., channel width). The result is an integrated transport in units of volume. Since the highly directional nature of the flow closely parallels the tidal period, near equal volume flows passed the current meter location, first in one direction, then in reverse.

Instantaneous peak transports are calculated similarly.

$$T_v = As$$

where T_v is instantaneous transport (in units of volume / time), A is again, cross sectional area, s is now the instantaneous peak speed. The computed transports are summarized in Table 7.

TABLE 7. VOLUMETRIC TRANSPORTS AT SELECTED LOCATIONS.

Station	(Ts) Volume per 1/2 cycle	(Tv) Peak transport
B14	1.2 x 10^7 m^3	900 m^3/sec
WCH	6.0 x 10^7 m^3	3400 m^3/sec
MN4	4.2 x 10^7 m^3	2700 m^3/sec
total	1.14 x 10^8 m^3	

The total average flow (T_s) equates to within an order of magnitude to the tidal prism of 1.15×10^8 cubic meters calculated from the volumetric analysis of the bay. It also equates to the total equivalent volume of the lagoon circulating around the island every 10 tidal cycles. This balance of flow between the principal boundaries of exchange may further indicate that the mean residence time of a water parcel within the lagoon may approach ten days.

V. CONCLUSIONS

As a first step in quantifying Johnston Atoll's physical oceanography, this study has identified the major components of current flow, categorized the circulation, and attempted to define the fundamental properties important in the circulation. An extensive effort to model the seiche resulted in reasonable numerical approximation to the bay's modal oscillation. The model generated seiche periods appear to agree with the observed oscillation of Johnston Atoll and seiche mode energy appears to be a significant contributor to the total flow. Oscillations occur at the diurnal (K1) and semi diurnal (M2) tidal frequencies, the inertial frequency (f) and its first harmonic, modulation of the M2 and inertial frequency, and the fundamental seiche mode. Several current records had consistent peak amplitudes clustered around the first two modes ($T_{1,0}$ = 140 and $T_{2,0}$ = 70 minutes) of free oscillation as derived by the three dimensional seiche model. The geographic positioning of model generated modal shapes closely resemble theoretical expectations. The placement of current meters and instrumentation for future measurements should be improved, benefiting from this knowledge.

The seiche models are sufficiently flexible to apply in many elliptic and circular Island lagoons and estuaries. They can also be readily adapted as a teaching aid or laboratory exercise.

Analysis of currents within the bay resulted in the identification and characterization of the total flow. Currents are, in general, highly polarized and phase locked to the diurnal tide with an offset of about 160 degrees. Simple volume transport calculations over an average tidal period indicate that approximately 10 percent of the volume of the bay circulates around the main island every tidal cycle. This concurs with rudimentary calculations of mean residence time within the lagoon of about ten days.

A bathymetric data base has been created that, along with the frequencies and amplitudes of the major energy sources, identified herein, should enable subsequent modelling of the greater circulation, diffusion properties and computations of the island's total energy budget. Additionally, this study has provided the groundwork from which to conduct a sediment transport study.

A. SUGGESTIONS FOR FUTURE RESEARCH

As this study only partly quantified the basic circulation, much more needs to be done. Concurrent wind and current meter data needs to be collected. Salinity, temperature and density measurements also need to be made. Placement of current meters along the principal nodes, and placement of pressure gauges along the anti-nodes and outside the confines of the lagoon are needed to quantify exchanges with the greater circulation. Once wind driven components of flow have been analyzed and combined with the major components of energy identified in this study, a model of the greater circulation incorporating density stratification,

frictional influences, and diffusion coefficients can be developed utilizing the elliptical coordinate system developed here. The potential also exists to apply the finite element model of tidal flow developed by Ninomiya and Onishi (1991) which would assist in quantifying the dispersion and diffusion properties of pollutants, critical to the proper management and risk assessment of the JACADS project.

LIST OF REFERENCES

Ames, W. F. , 1977: *Numerical Methods for Patrial Differential Equations,* Second Edition, Academic Press, Inc., San Diego, California, 349 pp.

Barcley, Richard A., 1972: Johnston Atoll's Wake, *Journal of Marine Research*, Vol 30(2) pp. 201-216.

Botha, J. F., 1983: *Fundamental Concepts in the Numerical solution of Differential Equations*, John Wiley and Sons, New York, New York,198 pp.

Defant, A. (1960) *Physical Oceanography; Vol II*, Pergamon Press, Oxford, England.

Department of the Navy publication,1993: From The Sea, *Preparing the Naval Service for the 21st Century,* U. S. Government printing Office.

Dougherty , E. R., 1990: *Probability and Statistics For the Engineering, Computing and Physical Sciences*, Prentice Hall, Englewood Cliffs, New Jersey,780 pp.

Garbow, B. S., et. al., 1977 Matrix Eigensystem Routines - EISPACK Guide Extension, Lecture Notes in Computer Science, Vol 51, Springer-Verlag, Philadelphia, Pennsylvania.

Goldstein, S., 1965: Modern Developments in Fluid Dynamics, Vols 1 & 2, New York Dover Publications,Inc., New York.

Hsu, H. P., 1884: *Applied Fourier Analysis,*.Harcourt Brace Jovanovich, San Diego California, 223 pp.

Ippen, A. T., et. al., 1966: *Estuary and Coastline Hydrdynamics*, Engineering Societies Monographs, McGraw-Hill Book Company, Inc., New York, New York.

Ninomiya, H. , Onishi, K., 1991: *Flow Analysis using a PC,* Computational Mechanics Publications, Southhampton, Boston, 190 pp.

MATLAB (4.0) Users Guide, Reference manual, and Signal Processing toolbox, 1992: The Mathworks, Inc., Natick, Massachusetts.

Micronautics, Inc., 1994: Tide 1, Tide Prediction Software for the IBM PC. Copywrite 1986 - 1994, Rockport, Maine.

Nelson, R. C. 1993: The Indirect Measurements of Currents over Offshore Reefs, *Applied Ocean Research,* 15, 183- 192

Pickard, G. L., Emery, W. J., 1990: *Descriptive Physical Oceanography,* 5th edition, Pergamon Press, Oxford.

Pond, S, Pickard, G. L., Andrews, J. C., Wolanski, E., 1989: *A reveiw of the Physical Oceanography of the Grear Barier reef*, Austrailian Institute of Marine Science.

Saab, Y., 1989: Numerical Solutions of Large Nonsymetric Eigenvalue Problems, *Computational Physics Comm.*, 53:71 - 90.

Saab, Y., 1991: Numerical Methods for Large Eigenvalue Problems, *Manchester University Press Series in Algorithms and Architectures for Advanced Scientific Computing*, Manchester University Press, New York, New York.

Smith, G. D., 1978: *Numerical Solution of Partial Differential Equations, Finite Difference Methods, Second Addition*, Clarendon Press, Oxford, Oxford University Press.

Spiegel , M. R. 1992: *Mathmatical Handbook of Formulas and Tables*, Schaums Outline Series, McGraw Hill Inc., New York, New York.

Werner, F, E., 1992: *Tidal Hydrodynamics, Quantitative Aspects,* Encyclopedia of Earth science, Vol 4

Wilson, B. W., Hendrickson , J. A., Kilmer, R. E., 1965: *Feasibility Study For A surge Action Model of Monterey Harbor, California,* U. S. Army Corps of Engineers Contract No DA-22-079-civeng-65-10, Science Engineering Associates, San Marino, California.

Wolanski, E., 1984: Island Wakes in Shallow Coastal Waters, , *Journal of Geophysical Research,* Vol 89 No C6, pp 10,553-10569, Nov 1984.

APPENDIX

BASIS FOR THE SEICHE MODELS

These programs are designed to compute the free modes of oscillation of bays and harbors of varying geometry. They are numerical solutions to the wave equation:

$$A\nabla_h^2\Phi + \frac{1}{g}\Phi_{tt} = 0$$

where the linearized free surface condition:

$$\Phi_t = -g\eta$$

has been applied, with an assumed solution of the form:

$$\eta = \eta(x,y)e^{(i\sigma t)}$$

Specifics of the derived equations can be found in Wilson (1965) and in the main body of this thesis.

There are 3 separate models. One for the two-dimensional approximation to basins of arbitrary geometry. One for three-dimensional approximations to

rectangular basins, and one for three dimensional approximations to basins of elliptical or circular symmetry. Input generating programs are provided to generate model input from simple geometric values of length, width and depth. The equations as written, expect input depth vectors (matrices) to be expressed as positive meters from MSL. In developing these models, it became apparent that accurate estimates of the periods of oscillation from the model are critically dependent upon the proper specification of the boundary condition. Therefor, the three-dimensional models allow specification of two types of boundary conditions, or a depth dependent combination of both types.

The type 0 boundary is the boundary where the free surface elevation is assumed to be 0 at the boundary. That is, $\eta = 0$ is applied at the entire boundary. This is the condition for a fully enclosed basin: for instance, a lake.

The type 1 boundary is a fully open basin where the velocity normal to the boundary is assumed to be zero. Here, $d\eta/dx = 0$ is applied at the entire boundary. This condition is not realistic for most basins, but is available for selection.

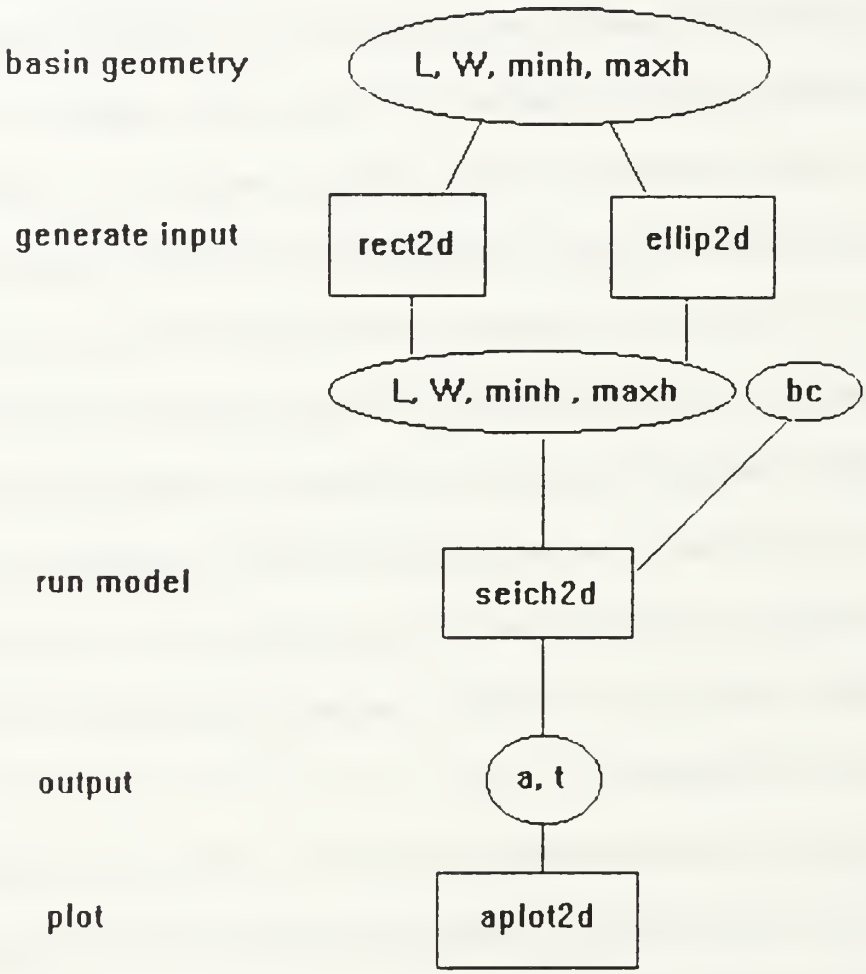
The type 2 Boundary is a depth dependent application of types 0 and 1. The criteria for selection of boundary conditions is based on whether the depth at a particular boundary point is less than the mean depth. If the depth at that point exceeds the mean, boundary condition 1 is applied. If the depth is less than the mean, boundary condition 0 is applied. This may not be optimum for some geometries; for example, a flat basin would default to boundary condition 0 fully (closed). Unless the depths at the boundaries are altered to trigger boundary

condition 1 at the open boundary, the modal shapes and the periods will not reflect the true nature of the oscillation. For this reason, if you specify a number for the boundary condition, other than 0, 1, or 2, (for instance setting bc=3) boundary condition 0 will be applied at depths less than or equal to 3 meters and boundary condition 1 will apply where depths are greater 3 meters. Thus, for a flat basin of 10 meters and open at one end, you would have to set the depths at the open boundary to a depth slightly greater than 10. (For example: 10.1). and specify bc=10 when you execute the model. The model will then apply boundary condition 0 at the three boundaries of depth 10 meters and boundary condition 1 at the open boundary of depth 10.1 meters. To specify the location of a known node line, set the depths at the location of the node line equal to zero. Some experimentation may be necessary to determine the proper mix of boundary conditions. Calculating the analytic solution for a given basin will serve as a base line for comparison to the numerical calculation. See Ippen (1965) or the main body of this thesis for the analytic solution to the equations.

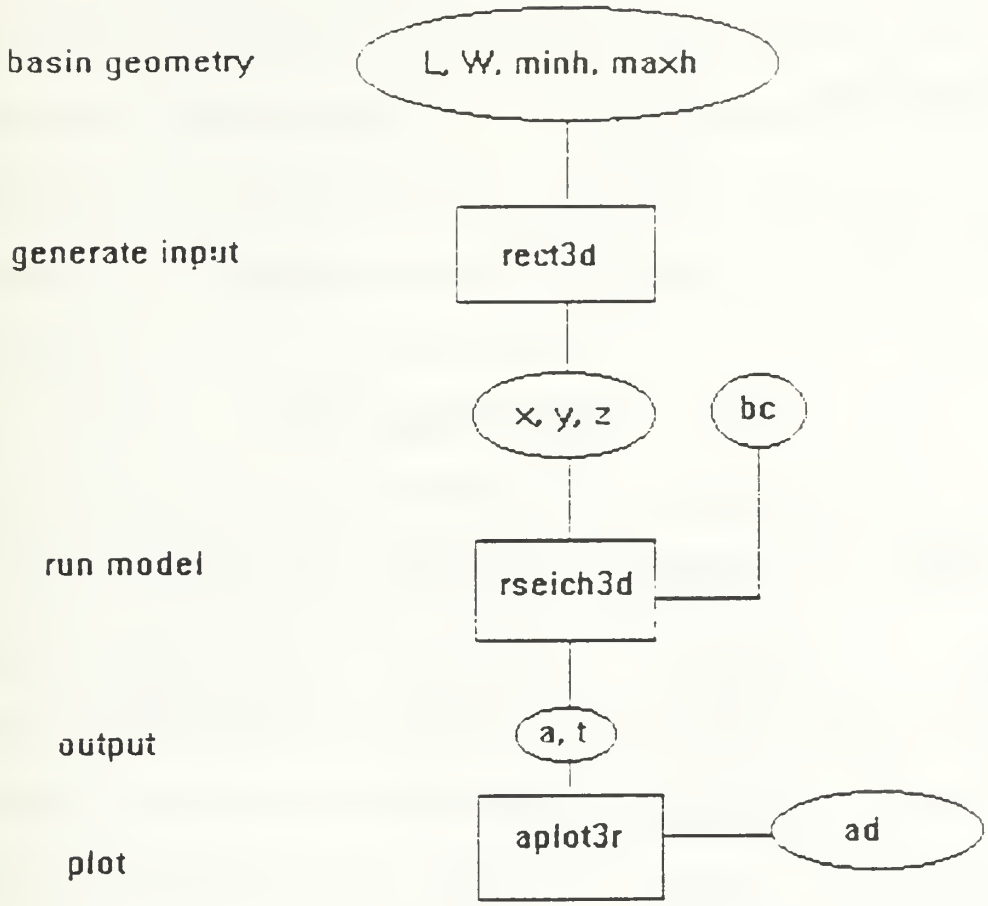
Flow diagrams are provided as reference for running the model and amplification of the individual model components is provided to clarify the specifics of each component. In addition, a short explanation of the individual component specifics is offered at the beginning of each program file. These programs are ascii text files and included in the accompanying diskette.

FLOW DIAGRAMS FOR EXECUTION OF THE SEICHE MODELS

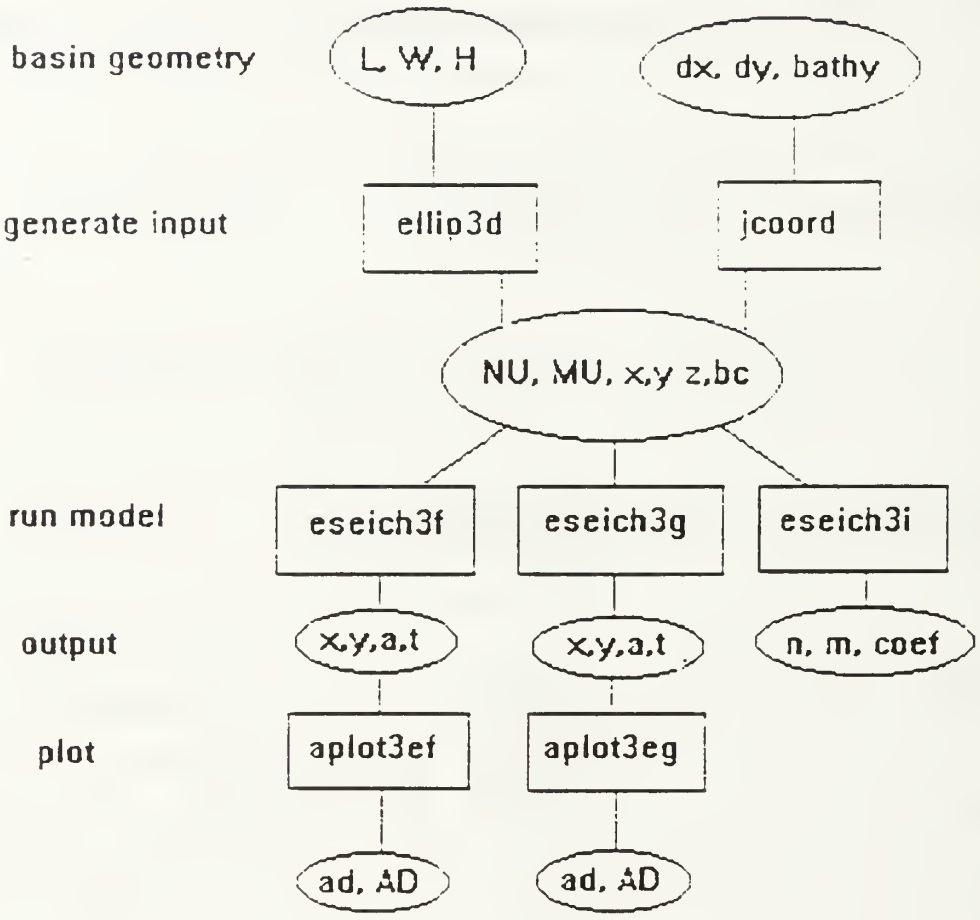
Variables are enclosed in ellipses, model components (functions) are enlosed in rectangles, and process steps are stated at the left margin.



THREE DIMENSIONAL RECTANGULAR MODEL:



THREE DIMENSIONAL MODEL IN ELLIPTIC COORDINATES:



MODEL DOCUMENTATION

The programs are designed to run while in the MATLAB (4.0) environment. MATLAB (4.0) is similar to FORTRAN and other programming languages. It is assumed here that the reader has a basic familiarity with MATLAB (4.0). Consult the user and reference manuals for assistance. The models and their sub components are designed in MATLAB (4.0) parlance as 'functions'. They all work in the form

[VAR1, ... , VARN] = functname(var1, ..., varn)

and are the MATLAB (4.0) equivalent of FORTRAN subroutines. The output variables are in brackets and the input variables are in parenthesis. Functions generic to the MATLAB (4.0) software are not discussed here. As with most MATLAB (4.0) functions, by typing " help functname" , a short description of the function's purpose, input and output variables is displayed. The functions described here are specialized functions created for the Johnston Atoll data set. Some may not function correctly in other applications, without some modification, though an attempt was made to make them general enough to be useful in other geometries. You may desire to design a script file to generate the necessary input prior to running the model, and call the individual functions from within the script. In the text of this discussion, when refering to variables are [bracketed], regardless of their status as input or output. Model components are **bold face** for clarification.

TWO-DIMENSIONAL MODEL

A. Generate input: Call the function:

[x,b,h]=ellip2d(L,W,maxh,minh,dx);

for a 2 dimensional elliptical basin, or

call the function:

[x,b,h]=rect2d(L,W,maxh,minh,dx);

to generate a 2 dimensional rectangular basin. Here, [L] = length, [W] = width, [maxh] = max depth and is applied at point 0. [minh] = min depth and is applied at point N. [dx] is the distance between discrete points from zero to [L]. You may also generate the input yourself, or load files with data from another source. The model will except arbitrary widths, as long as they represent the cross-sectional distance along discrete points from 0 to L. [x] is the vector [0:dx:L], [b] is basin widths corresponding to the points in [x], and [h] is the associated depths along [x].

B. Run the model: Call the function:

[a,t]=seiche2d(x,b,h,c);

The function expects input in the form generated by **rect2d** or **ellip2d**. Three column vectors [x, y, z]; [x], a distance vector -- Each entry in the vector is distance from the origin. [b], is the basin width at the distance specified in [x]. [h] is the vector of depths associated with [x, b]. If you type a fourth non-string entry as input [c], a plot is generated of the coefficient matrix non-zero element locations, otherwise no plot is generated. The Boundary condition is automatically

applied. Boundary condition zero; if z(N)< **mean(z)** or boundary condition one; if z(N) >=**mean(z)**.

C. Plot output: Call the function:

[AD]=aplot2d(x,h,a,t);

The output from **seiche2d**, [a, t] is plotted using this function. You can type "help **seiche2d**" for a description.

THREE DIMENSIONAL MODEL IN RECTANGULAR COORDINATES.

A. Generate input: Call the function:

[x,y,z]=rect3d(L,W,Hmax,Hmin,dx,dy);

where [L] = length ,[W] = width, [Hmax] = max depth , [Hmin] = min depth. [dx,dy] are the resolution desired in forming [x, y]. This function generates [x, y, z] matrices where rows of [x] are the vector [0:dx:L], and columns of [y] are the vector [0:dy:W]. [z] is uniformly sloping from [Hmax] at x = 0 to [Hmin] at [x] = [L]. If [Hmin] = [Hmax] a flat basin is formed. The output variables [x, y, z] are matrices, representing the x, y, z coordinate triple for length, width, and depth; x(i,j), y(i,j) and z(i,j). which represent the [x, y, z] values of the point (i,j). See the documentation concerning the MATLAB (4.0) function "**meshgrid**" on how to generate these matrices from other data.

B. Run the model Call the function:

[ADA, T] = rseich3d(x,y,z);

This function expects input in the form generated by rect3d. [x, y] can be of varying resolution, but the program code requires x and y to be of the form a(i) = i*Δa for all a(i). That is, [x, y] must be equally spaced. The limitation does not apply to [z]. [z] can be anything, except no provision was made to expect interior boundaries: entries in [z] must be non-zero.

NOTE : Computer memory limitations may require limiting the matrix size to [N, M] = [20, 20].

C. Plot the output: Call the function:

[ad]=aplot3r(x,y,z,ADA,T);

You will be asked which mode to plot; only one integer value is acceptable. The number you type corresponds to the mode number; 1 for the first mode, 2 for the second, etc. The second plot is a contour plot . You will be asked to label the contours by clicking on them with the mouse.

THREE DIMENSIONAL MODEL IN ELLIPTICAL COORDINATES

This follows the same pattern, but gets more complicated; Therefor, default data sets: **x20.dat, y20.dat, z20.dat, nu20.dat, mu20.dat, cf20.dat** are provided. In addition, the main data set for the Johnston Island bathymetry is **jb250m.dat**. The three elliptic model functions call several other specialized functions that should be transparent to you. These primarily involve transformations of grid coordinates, etc. For example, **ij2latlon** transforms an i, j grid to latitude and longitude specifically for the Johnston Island, much the same way as the MATLAB (4.0) **cart2pol** function transforms cartesian coordinates to polar. Modification of this function will be necessary for other latitudes and longitudes. Similarly, **xy2latlon** does the same for two matrices [x, y] if they are coordinate pairs and represent distances in meters.

A. Generate input:

Load data files containing the variables[NU, MU, x, y, z, Cf], or, call the function:

[NU,MU,xp,yp,z,cf]=ellip3d(L,W,H);

This function works the same as rect3d and generates the necessary matrices that are required for running **eseich3f.m** and **eseich3g.m. Ellip3d** only generates a flat bottom of depth [H]. You may also generate matrices from the Johnston Atoll Data sets by calling the function:

[NU,MU,xp1,yp1,Z,Cf]=jcoord(dx,dy,bathy);

This function expects [dx, dy] and [bathy], an input matrix in cartesian space of bathymetry; something like jb250m.dat. [dx, dy] represent the resolution of [bathy]. The function generates an elliptic coordinate system mapped to points chosen from [bathy] and interpolates [Z] from [bathy] into the coordinates formed this way. It then generates the other variables needed to run the model. This is a rather complicated function, but it is menu driven. You will be asked to enter a resolution, and graphically input boundary points from the plots generated, then pick the center point from which the elliptical coordinates are interpolated. There is also a feature that asks you if you want the output matrices [xp1, yp1] to be in Latitude/longitude or meters. [Cf] is the three element vector containing the Center to focus distance, the semi-major axis, and the semi-minor axis of the ellipse you have chosen. Only the first entry of [Cf] is needed to run the model.

NOTE: **Jcoord** prompts for EVEN size Matrices, but, the elliptical model is designed for ODD size input matrices, where **[n,m]=size(NU)** returns [n] rows and [m] columns, the size of [NU]. If [n, m] are not odd, the model will give erroneous output and/or won't run at all. If you input even integers into **jcoord or ellip3d** when prompted, it will automatically give you the next higher odd size matrices, precisely what the model needs.

You have three choices of the elliptic models. All work the same way, but generate output in slightly different ways.

B. Run The Model: Call the function:

[a,t]=eseich3f(NU,MU,z,cf,bc);

This version uses cartesian coordinates at the foci. It calculates the eigenvalues and eigenvectors using MATLAB (4.0). [cf] is the first entry in the matrix [Cf] generated with **ellip3d.m**, and **jcoord.m**, or:

Call the function:

[a,t]=ellip3g(NU,MU,z,cf,bc)

This version eliminates the singularity condition of the foci, and renumbers the grid. It also calculates the eigenvectors with MATLAB (4.0), and seems to work well. You may also:

Call the function:

[n,m,coef] = ellip3i(NU,MU,z,cf,bc)

This version calculates only the (n x m) x (n x m) coefficient matrix for solving with another eigenvalue solver. Try this version only if you don't feel MATLAB (4.0) is giving you what you need. You will have to unravel the eigenvalues and eigenvectors from whatever eigenvalue solver you use to arrive at meaningful numbers. Also, because the way in which the coefficient matrix is formed, if you wish to plot, you will have to reconstruct the eigenvectors in the same manner that the plot function **aplot3eg** does, keeping track of the indices. This may be more trouble than its worth.

C. Plot the output: If you ran **ellip3g**, call the function:

[ad,AD]=aplot3eg(x,y,a,t)

This function takes the m matrices [y, y] to be of the form generated by **jcoord** or **ellip3d** and plots the output from **eseich3g** only !

If you ran **eseich3f,** call the function:

[ad,AD]=aplot3ef(x,y,a,t)

Like **aplot3eg**, this one only plots output generated by **eseich3f.** These plot functions reconstruct the eigenvectors in their index locations, and reformat the column of [a] corresponding to the mode that you select. All work simply, same as **aplot3r** does for the rectangular model. You can call **rplot3d, aplot3eg** and **aplot3ef** repeatedly, to plot modes of your choice, selecting different modes, without rerunning the model. **AD** and **ad**, are the reformed matrices of the column of [a]. Lowercase [ad] is in elliptic coordinates, Uppercase [AD] is in cartesian coordinates. No provision is made to plot output from **ellip3i.**

EXAMPLE MODEL RUN

As an example, these statements written in a script file, or in the MATLAB (4.0) environment will generate input, run the three dimensional elliptical model, version g, and plot the output for a basin of flat bathymetry, that is 10000 m long, 8000 m wide and 20 m deep at a ten by ten resolution:

The >> indicate the MATLAB (4.0) prompt.

```
>> L=10000;
>>W=8000;
>> H=20;
>> bc=2; % apply a boundary based on depth
>> [NU10,MU10,x10,y10,z10,cf10] = ellip3d(L,W,H);
inter your resolution [even integers] : [10,10]
>> [a10,t10] = eseich3g(NU10,MU10,z10,cf10(1),bc);
>> [ad10,AD10] = aplot3eg(x10,y10,a10,t10);
mode to plot? : 1
```

LISTINGS OF PROGRAM FILES.

```
% contents.m
% contents of seiche model disk
% mfiles and their discriptions

% APLOT2D;   function [p]=aplot2d(x,h,a,t);

%          plot output from seiche2d.m

% APLOT3EF;  function [ad1,ADII] = aplot3ef(xpl,ypl,ada,t)

%       meshplots output from eseiche3f.m, 3d representation of
%       A GENERIC ELLIPTICAL BASIN

%APLOT3EG;   function [ad1,ADII] = aplot3eg(xpl,ypl,ada,t)

%       meshplots output from ellip3g.m, 3d representation of bay
%       uses (xpl,ypl,ada,t) generated from eseich3g.m

%ESEICH3G;   function [ada,t]=eseich3g(nu,mu,z,cf,bc)

%       matlab function to determine the 3 dIMensional modes of
%       oscillation of an ELLIPTICAL basin under gravity.

%ESEICH3F;   function [ada,t]=eseich3f(nu,mu,z,cf,bc)

%       matlab function to determine the 3 dIMensional modes of
%       oscillation of an ELLIPTICAL basin under gravity.

%ELLIP3I;    function [coef,zbar]=ellip3i(nu,mu,z,cf)

 %      matlab function to determine the coef matrix used to determine
%       the 3 dIMensional modes of
 %      oscillation of an ELLIPTICAL basin under gravity.

%ELLIP2D;    function [x,b,h]=ellip2d(L,W,maxH,minH,dx)

%       makes a data set for seich2d.m with an
 %      elliptical bay of  flat bottom or uniformly sloping

%ELLIP3D;    function [NU,MU,xp,yp,z,cf]=ellip3d(L,W,H);

%       generates an elliptical coord system for
```

```
%       use in eseich3g.m or eseich3f.m

%JCOORD;   function [NU,MU,xp1,yp1,Z,Cf]=jcoord(dx,dy,bathy)

%       interpolates and tranlates a subset of
%       the input matrix (bathy) into  elliptical cooordintes

%IJ2LATLN;  function [lon,lat]=ij2ltln(n,m,dx,dy);

%       makes an [i,j] grid and converts it to long lat
%       at 16 geg n lat

%JPLOT3EG;   function [ad1,ADII] = jplot3eg(xpl,ypl,ada,t)

%       meshplots output from ellip3g.m, 3d representation
%       of seiche modes
%               SPECIFICALLY FOR JOHNSTON ATOLL

%JPLOT3EF;  function [ad1,ADII] = jplot3ef(xpl,ypl,ada,t)

%       meshplots output from eseiche3f.m 3d representation of
%            of Johnston Atoll seiche modes
%       uses (xpl,ypl,ada,t) generated from eseich3f.m

%RECT2D;     function [x,y,z]=rect2d(L,W,Hmax,Hmin,dx);

%       forms a rectangular basin for use in seiche2d.m

%RECT3D;   function [x,y,z]=rect3d(L,W,Hmax,Hmin)

%       generates a rectangular basin to run rseich3d.m

%XY2LATLON;  function [lon,lat]=xy2ltln(x,y);

%       takes coordinate vectors or matrices
%       in meters and and converts them to long lat
%       at 16.4 deg n lat

%SEICHE2D;  function [a,t]=seiche2d(x,b,h,c)

%       matlab mfile to determine the 2 dimensionsl modes of
%       oscillation of a bay under gravity
```

```
function [p]=aplot2d(x,h,a,t);
% function [p]=aplot2d(x,h,a,t);
% mfile to plot output from bayfrq.m
% input is [x] dist vector
% [h] depth vector
% [a] modes to plot
% [t] periods to plot in seconds
% output is a graph
disp('aplot2d')
H=mean(h);
figure
p=plot(x,a(:,1),x,a(:,2),x,a(:,3),x,a(:,4),x,a(:,5))
  xlabel('Distance')
  ylabel('Relative Amplitude')
title('First 5 2d modes ')
text(x(2),max(a(:,1)),['1st 5 Periods (min) = ',num2str(t(1)/60),' ',...
num2str(t(2)/60),' ',num2str(t(3)/60),' ',...
num2str(t(4)/60),' ',num2str(t(5)/60)])
text(x(2),max(a(:,1))*.75,['Hbar= ',num2str(H)])
hold off
```

```
function [ad1,ADII] = aplot3ef(xpl,ypl,ada,t)
%function [ad1,ADII] = aplot3ef(xpl,ypl,ada,t)
% meshplots output from eseiche3f.m 3d representation of
%  A GENERIC ELLIPTICAL BASIN
% uses (xpl,ypl,ada,t) generated from eseich3f.m
% loads jb250m.dat, calls ij2latln.m
%
% contours ada after tranforming to an xy uniform
% grid
disp('aplot3ef.m')


              %0:dx:(m-1)*dx;
              %0:dy:(n-1)*dy;
              %[lon,lat]=meshgrid(lon,lat);


[IM,JM]=size(xpl);  % size of matrix


mode=input('mode to plot?');

ad=ada(:,mode);

I=1:IM*JM;

II=reshape(I,JM,IM)';


    f1 = II(1,1:(JM+1)/2-1);

    fIM = II(IM,(JM+1)/2+1:JM);

  I(fIM) = [];

   I(f1) = [];

I=I';


ad1(I)=ad;
```

```
AD1=ad;

 x=xpl'; y=ypl';

 x=reshape(x,IM*JM,1);
 y=reshape(y,IM*JM,1);


x(fIM)=[];
x(f1)=[];

y(fIM)=[];
 y(f1)=[];



% get rid of duplicated points


   ad1(length(ad1)+1:IM*JM)=zeros(size(fIM));

   ad1=reshape(ad1,JM,IM)';


   nn=zeros(size(f1));

     for i=1:length(nn)

         nn(i)=nan;
     end


ad1(1,1:(JM+1)/2-1)= fliplr(ad1(1,(JM+1)/2+1:JM));
ad1(IM,(JM+1)/2+1:JM)=fliplr(ad1(IM,1:(JM+1)/2-1));
%ad1(1,(JM+1)/2)=nan;
%ad1(IM,(JM+1)/2)=nan;

mn=menu(' Mesh plot of the mode ?','Yes','No')
if mn==1
figure;

    mesh(xpl,ypl,ad1);axis('equal');
    title(['Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
```

```
    ' (min)  ellip Basin'])
    xlabel(' distance ')
    ylabel(' distance ')

end


% this part contours ada by transforming
% x,y,ada to an evenly spaced grid
% and figures out a way to contour ada
% from eseich3f.m



     XPo=min(x);
   Xpmax=max(x);
     dxp=(Xpmax-XPo)/(15);
     xpp=XPo:dxp:Xpmax;
     YPo=min(y);
   Ypmax=max(y);
     dyp=dxp; % (Ypmax-YPo)/(10);
     ypp=YPo:dyp:Ypmax;


[XPP,YPP]=meshgrid(xpp,ypp); % uniform grid


     ADII=griddata(x,y,AD1,XPP,YPP);



% contour vector
      dv=(max(max(AD1))-min(min(AD1)))/5;
      vo=min(min(AD1));
      vmax=max(max(AD1));
      v=vo:dv:vmax;

mn=menu('Contour plot of the mode?','Yes','No')
if mn==1
figure;
     plot(xpl(:,1),ypl(:,1),'w',xpl(:,JM),ypl(:,JM),'w');

     hold on
```

```
    c=contour(XPP,YPP,ADII,[0,v],'r--');axis('equal');
    clabel(c,'manual')
    title(['Contours of Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
      ' (min)  ellip Basin '])
    xlabel('distance')
    ylabel('distance')
      axis('equal')
hold off
end
```

```
function [ad1,ADII] =aplot3eg(xpl,ypl,ada,t)
%function [ad1,ADII] = aplot3eg(xpl,ypl,ada,t)
% meshplots output from ellip3g.m 3d representation of bay
% uses (xpl,ypl,ada,t) generated from eseich3g.m
%
% contours ada after tranforming to an xy uniform
% grid
disp('aplot3eg.m')

    xaxl='Meters';
    yaxl='Meters';

mlon=xpl;
mlat=ypl;

[IM,JM]=size(mlon);  % size of matrix

mode=input('mode to plot?');

ad=ada(:,mode);

I=1:IM*JM;

II=reshape(I,JM,IM)';


    f1 = II(1,1:(JM+1)/2);

    fIM = II(IM,(JM+1)/2:JM);


  I(fIM) = [];

   I(f1) = [];

I=I';


ad1(I)=ad;
AD1=ad;
 x=mlon'; y=mlat';
 x=reshape(x,IM*JM,1);
 y=reshape(y,IM*JM,1);
```

```
x(flM)=[];
x(f1)=[];

y(flM)=[];
 y(f1)=[];



% get rid of duplicated points

%[n,m]=size(l);

%  X=reshape(x,n,m);
%  Y=reshape(y,n,m);



ad1(length(ad1)+1:lM*JM)=zeros(size(flM));

ad1=reshape(ad1,JM,lM)';


   nn=zeros(size(f1));

     for i=1:length(nn)

         nn(i)=nan;
     end

%ad1(:,JM)=ad1(:,JM-1)
ad1(1,1:(JM+1)/2)= fliplr(ad1(1,(JM+1)/2:JM));
ad1(lM,(JM+1)/2:JM)=fliplr(ad1(lM,1:(JM+1)/2));
ad1(1,(JM+1)/2)=nan;
ad1(lM,(JM+1)/2)=nan;


ms=menu('mesh plot of the mode ?','Yes','No');
if ms==1
```

```
   mesh(mlon,mlat,ad1);axis('equal');
   title(['Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
   ' (min)'])
   xlabel(xaxl)
   ylabel(yaxl)
   zlabel('Nondimensional Elevation')

end
keyboard

% this part contours eta by transforming
% x,y,ada to an evenly spaced grid
% figure out a way to contour ada
% from eseich3g.m



    XPo=min(x);
  Xpmax=max(x);
    dxp=(Xpmax-XPo)/(15);
    xpp=XPo:dxp:Xpmax;
    YPo=min(y);
  Ypmax=max(y);
    dyp=dxp; % (Ypmax-YPo)/(10);
    ypp=YPo:dyp:Ypmax;


[XPP,YPP]=meshgrid(xpp,ypp); % uniform grid


    ADII=griddata(x,y,AD1,XPP,YPP);



% contour vector
     dv=(max(max(AD1))-min(min(AD1)))/6;
     vo=min(min(AD1));
     vmax=max(max(AD1));
     v=vo+dv:dv:vmax;

    v=[0,vo,v,vmax];

ms=menu('contour plot of the mode ?','Yes','No');
```

```
if ms==1


    plot(mlon(:,1),mlat(:,1),'w',mlon(:,JM),mlat(:,JM),'w');

    hold on

    c=contour(XPP,YPP,ADII,[0,v],'r--');axis('equal');
    clabel(c,'manual')
    title(['Contours of Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
      ' (min)'])
  xlabel(xaxl)
  ylabel(yaxl)

    axis('equal')

hold off
end
```

```
function [ad]=aplot3r(x,y,z,ada,t)

% function [ad]=aplot3r(x,y,z,ada,t)
% meshplots ada, the output from rseich3.m
% input is x,y,z matrices, and a,t output from rseich3
% output is a matrix of the eigenvector a(:,mode)
% where mode represents the column of a;

[IM,JM]=size(z);
disp('aplot3r')
m=input('mode to plot = ? ')
Z=mean(mean(z));
ad=(reshape(ada(:,m),JM,IM))';
dad=(max(ada(:,m))-min(ada(:,m)))/6;
mesh(x,y,ad)
axis('equal')
%-(min(dd)-1);
  title(['Mode ',num2str(m),' T= ',num2str(t(m)/60),...
         ' (min) Rect Basin, H =',num2str(Z)])
  xlabel('meters ')
  ylabel('meters ')
keyboard
plot(x(:,1),y(:,1),'w',x(1,:),y(1,:),'w',x(:,JM),y(:,JM),'w',...
x(IM,:),y(IM,:),'w')
hold on
axis('equal')
c=contour(x,y,ad,[min(min(ad)):dad:max(max(ad))]);
clabel(c,'manual')
 title(['Mode ',num2str(m),' T= ',num2str(t(m)/60),...
         ' (min) Rect Basin, H =',num2str(Z)])
  xlabel('meters ')
  ylabel('meters ')
hold off
```

```
function [x,b,h]=ellip2d(L,W,maxH,minH,dx)
% function [x,b,h]=ellip2d(L,W,maxH,minH,dx)
%
% makes a data set for seich2d.m with an
% elliptical bay of  flat bottom or uniformly sloping
%
% output is three column vectors [x,b,h]
% representing length , widths, depths
% input :  L = length of basin
%            W = max width of basin
%     maxH = max depth
%     minH = min depth
%          dx = distance between points
%  if dx not specified,  default dx is 250 meters;
%  if maxH = minH a flat basin is generated
% otherwise MaxH is depth at x=0
% and minH is depth at x=L;
%
if nargin<=4
dx=250;
else
end
x=-L/2:dx:L/2;
n=length(x);
smajor=(L/2);
sminor=W/2;
    ysqd=(1-(x.^2/smajor^2))*sminor^2;
b=sqrt(ysqd);
b=2*b;
b(1)=dx;
b(n)=dx; % discard the limits b ==>0
% H same length as b
if maxH== minH;
  h=zeros(size(b));
  h=h+maxH;
else
 dh=-(maxH-minH)/(n-1);
  h=maxH:dh:minH;
end
x=x';
b=b';
h=h';
```

```
function [NU,MU,xp,yp,z,cf]=ellip3d(L,W,H);
%function [NU,MU,xp,yp,z,cf]=ellip3d(L,W,H);
%
% generates an elliptical coord system for
% use in eseich3g.m or eseich3f.m
% L= length, W=width,H=depth, all scalors
% screen input asks for specification of resolution
% depth is uniform except at boundaries
% where it is H/2 along one edge
cf = sqrt((L/2)^2-(W/2)^2) %center to focus dist
B=W/2;
A=L/2;

Mu=asinh(B/cf)

res=input('specify your resolution [Nu(IM rows),mu(JM columns)] even integers
');
dnu=pi/res(1);
Nu=pi;
dmu=Mu/res(2)*2;
u=0:dmu:Mu;
u=[-fliplr(u(2:length(u))),u];

% (columns of equal radius);

nu=-Nu:dnu:0;
IM=length(nu);
%  (rows of equal radius)

JM=length(u);

[MU,NU]=meshgrid(u,nu);

xp =cf*cosh(MU).*cos(NU);
yp =cf*sinh(MU).*sin(NU);

 z=zeros(size(MU));
 z=z+H;
 z(:,JM)=z(:,JM)-H/2;
cf=[cf,A,B]
```

```
function [coef,zbar]=ellip3i(nu,mu,z,cf,bc)

% ellip3i;
% function [coef,zbar]=ellip3f(nu,mu,z,cf,bc)
% matlab function to determine the 3 dIMensional modes of
% oscillation of an ELLIPTICAL basin under gravity.
%
% USES RECTANGULAR COORDINATES AT THE FOCI
% calls the mfile eig.m to decompose an JM*IM coef matrix
% input is specifed from data files
% named zi.dat, nu.dat, mu.dat, cfm.dat created by first running
% mfile xcoordb.m
% input required is z (matrix) and mu (matrix) grid point spacing
% nu (matrix) grid spacing (radians) and
% cf, center to focus dist in meters.
% all measurements in meters
% parameters are: g=9.81 gravity constant
% output is[coef] to be solved with another eigenvalue solver such as eigs.
%
%
%  coef is the coef matrix formed this way,
% Cnr is the condition nr of
% the coeficient matrix: cnr=cond(coef)
%

      [IM,JM]=size(mu);

      dnu=abs(nu(2,1)-nu(1,1))

      dmu=abs(mu(1,2)-mu(1,1))

      a=dnu^2;
      b=dmu^2;

      zbar = mean(mean(z)); %mean(mean(z));

    % verify all depths greater than = 0.0

       for j=1:JM;
          for i=1:IM;
              if z(i,j)<=0.0
                 z(i,j)=0.0;
              end
```

```
        end
      end

end


z = z/zbar;

o = mean(mean(z))
% delete the duplicated coordinate points

   nz=z(1,1:(JM+1)/2-1);

        for i=1:length(nz);

            nz(i)=nan;

        end

    z(1,1:(JM+1)/2-1)=nz;


    z(IM,(JM+1)/2+1:JM)=nz;

    smu=sinh(mu);

    snu=sin(nu);

    smu=smu.^2;

    snu=snu.^2;

    x=cf/cf*cosh(mu).*cos(nu);

    y=cf/cf*sinh(mu).*sin(nu);

      hsqd=smu+snu;
% remove the duplicated points and the singularity of hsqd:

        hsqd(1,1:(JM+1)/2-1)=nz;
        hsqd(IM,(JM+1)/2+1:JM)=nz;
        hsqd(1,(JM+1)/2)=nan;
        hsqd(IM,(JM+1)/2)=nan;
```

```
          hsqd=1./hsqd;

      g = 9.81;

   % matrix of indices

       I = 1:JM*IM;

       I = reshape(I,JM,IM)';

%%%%%%%% eigenvalue equations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

coef=zeros(IM*JM);

% Interior pts

    for j=2:JM-1
     for  i=3:IM-2

      coef(I(i,j),I(i,j))    = -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
      coef(I(i,j),I(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
      coef(I(i,j),I(i-1,j)) = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));
      coef(I(i,j),I(i,j+1)) = hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
      coef(I(i,j),I(i,j-1)) = hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

     end
   end

i=2;
    for j=2:(JM+1)/2-1

      coef(I(i,j),I(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,JM+1-j)+4*z(i,j));
      coef(I(i,j),I(i-1,JM+1-j)) = hsqd(i,j)*1/(4*a)*(z(i-1,JM+1-j)-z(i+1,j)+4*z(i,j));

    end
i=2;
j=(JM+1)/2;

     coef(I(i,j),I(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
     coef(I(i,j),I(i-1,j)) = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));
```

```
   for j=(JM+1)/2+1:JM-1

      coef(l(i,j),l(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
      coef(l(i,j),l(i-1,j)) = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

   end

i=2;

    for j=2:JM-1

      coef(l(i,j),l(i,j))   = -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
      coef(l(i,j),l(i,j+1)) = hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
      coef(l(i,j),l(i,j-1)) = hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

    end

%*********************************************************************

 i=1;

    for j=(JM+1)/2+1:JM-1

        % nu derivitive

    coef(l(i,j),l(i,j))     =   -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
    coef(l(i,j),l(i+1,j))   =    hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i+1,JM+1-j)+4*z(i,j));
    coef(l(i,j),l(i+1,JM+1-j)) = hsqd(i,j)*1/(4*a)*(z(i+1,JM+1-j)-z(i+1,j)+4*z(i,j));

        % mu derivitive

     coef(l(i,j),l(i,j+1)) =     hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
     coef(l(i,j),l(i,j-1)) =    hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

    end

% ***************** focus in rect coord ********************************


dx=abs(x(1,(JM+1)/2+1)-x(2,(JM+1)/2))/2;
dy=abs(y(2,(JM+1)/2-1)-y(2,(JM+1)/2+1))/2;

i=1;
```

```
j=(JM+1)/2;

   coef(l(i,j),l(i,j))   = -(2*z(i,j)/dx^2)-(2*z(i,j)/dy^2);
   coef(l(i,j),l(i+1,j)) = 1/(4*dx^2)*(z(i+1,j)-z(i,j+1))+z(i,j)/(dx^2);
   coef(l(i,j),l(i,j+1)) = 1/(4*dx^2)*(z(i,j+1)-z(i+1,j))+z(i,j)/(dx^2);
   coef(l(i,j),l(i+1,j-1)) = 1/(4*dy^2)*(z(i+1,j-1)-z(i+1,j+1))+z(i,j)/(dy^2);
   coef(l(i,j),l(i+1,j+1)) = 1/(4*dy^2)*(z(i+1,j+1)-z(i+1,j-1))+z(i,j)/(dx^2);


%*************************************************************************

 i=IM-1;

   for j=2:(JM+1)/2-1

       % nu derivitive

    coef(l(i,j),l(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
    coef(l(i,j),l(i-1,j)) = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

  end

  for j=(JM+1)/2+1:JM-1

          % nu derivitive

     coef(l(i,j),l(i-1,j))       = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,JM+1-j)+4*z(i,j));
     coef(l(i,j),l(i+1,JM+1-j)) = hsqd(i,j)*1/(4*a)*(z(i+1,JM+1-j)-z(i-1,j)+4*z(i,j));

   end

i=IM-1;
j=(JM+1)/2;

% nu derivitive

     coef(l(i,j),l(i+1,j))    =     hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
    coef(l(i,j),l(i-1,j))  =     hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

  for j=2:JM-1

     coef(l(i,j),l(i,j))     =   -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
```

```
      % mu derivitive

     coef(I(i,j),I(i,j+1)) =      hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
     coef(I(i,j),I(i,j-1)) =     hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

  end


%************************************************************************

 i=IM;

  for j=2:(JM+1)/2-1            % nu derivitive

    coef(I(i,j),I(i,j))     =   -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));

    coef(I(i,j),I(i-1,j))   =    hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i-1,JM+1-j)+4*z(i,j));
   coef(I(i,j),I(i-1,JM+1-j))=  hsqd(i,j)*1/(4*a)*(z(i-1,JM+1-j)-z(i-1,j)+4*z(i,j));

        % mu derivitive

     coef(I(i,j),I(i,j+1)) =      hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
    coef(I(i,j),I(i,j-1)) =      hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

  end


%**************** focus in rectangular coordinates
*********************************************

i=IM;
j=(JM+1)/2;

   coef(I(i,j),I(i,j))   = -(2*z(i,j)/dx^2)-(2*z(i,j)/dy^2);
   coef(I(i,j),I(i-1,j-1)) = 1/(4*dy^2)*(z(i-1,j-1)-z(i-1,j+1))+z(i,j)/dy^2;
   coef(I(i,j),I(i-1,j+1)) = 1/(4*dy^2)*(z(i-1,j+1)-z(i-1,j-1))+z(i,j)/dy^2;
   coef(I(i,j),I(i-1,j)) = 1/(4*dx^2)*(z(i-1,j)-z(i,j-1))+z(i,j)/dx^2;
   coef(I(i,j),I(i,j-1)) = 1/(4*dx^2)*(z(i,j-1)-z(i-1,j))+z(i,j)/dx^2;


%*****************************************************************************

% interior boundary of the island
```

```
[n,m]=find(z==0);

 for i=1:length(n)

    if n(i) > 2 & n(i) < IM-2 & m(i) > 2 & m(i) < JM-1

        coef(I(n(i),m(i)),I(n(i)-1,m(i)))=0;
        coef(I(n(i),m(i)),I(n(i)+1,m(i)))=0;
        coef(I(n(i),m(i)),I(n(i),m(i)-1))=0;
        coef(I(n(i),m(i)),I(n(i),m(i)+1))=0;

     end
 end


%%%%%%%%%%% BOUNDARY CONDITION
****************************************

 if bc==1

        j=JM;

   for i=1:IM-1;

      % at mu=MU backward differencing in mu

      coef(I(i,JM),I(i,JM))      =  hsqd(i,j)*3/(2*dmu);
      coef(I(i,JM),I(i,JM-1))    = - hsqd(i,j)*4/(2*dmu);
      coef(I(i,JM),I(i,JM-2))    =   hsqd(i,j)*1/(2*dmu);

    end

         j=1;

    % forward differencing in mu

   for i=2:IM

      coef(I(i,1),I(i,1))      =   - hsqd(i,j)*3/(2*dmu);
      coef(I(i,1),I(i,2))      =   + hsqd(i,j)*4/(2*dmu);
      coef(I(i,1),I(i,3))      =   - hsqd(i,j)*1/(2*dmu);
```

```
   end

elseif bc==2


%****************** mixed derivitve case ******************************************


   % o = zbar;  % 'min depth to apply apply dn/dmu=0 ')
   % o = o/zbar;

% at mu=MU backward differencing in mu

j=JM;
      for i=1:IM-1

         if z(i,j) >= o

            coef(I(i,j),I(i,j))     =   hsqd(i,j)*3/(2*dmu);
            coef(I(i,j),I(i,j-1))   = - hsqd(i,j)*4/(2*dmu);
            coef(I(i,j),I(i,j-2))   =   hsqd(i,j)*1/(2*dmu);

         end
      end



 % forward differencing in mu

j=1;

       for i=2:IM

          if z(i,j) >= o

               coef(I(i,j),I(i,j))    =   - hsqd(i,j)*3/(2*dmu);
               coef(I(i,j),I(i,j+1))  =   + hsqd(i,j)*4/(2*dmu);
               coef(I(i,j),I(i,j+2))  =   - hsqd(i,j)*1/(2*dmu);

          end
       end

end
```

```
%[i,j]=find(coef);

%figure;plot(i,j,'.');axis('ij')

%title('coef without deletion of rows')

% delete unused rows and columns

    f1=I(1,1:(JM+1)/2-1)

    fIM=I(IM,(JM+1)/2+1:JM)

        coef(:,fIM)=[];
        coef(fIM,:)=[];

        coef(:,f1)=[];
        coef(f1,:)=[];


%[i,j]=find(coef);

%figure;plot(i,j,'.');axis('ij')
%   title(' coef matrix with deleted rows')

disp('coef is size')

disp(size(coef))
%keyboard
cnr=cond(coef);
disp(['cond = ',num2str(cnr)])

break

disp('solving coef using matlab eig.m')


[a,l] = eig(coef);  % matlab eigen vector, eigen value function


  %***** unravel lamda squared  and ada **************
```

```
function [ada,t]=eseich3f(nu,mu,z,cf,bc)

% function [ada,t]=eseich3f(nu,mu,z,cf,bc)
% matlab function to determine the 3 dimensional modes of
% oscillation of an ELLIPTICAL basin under gravity.
%
%   USES RECTANGULAR COORDINATES AT THE FOCI
% calls the mfile eig.m to decompose an JM*IM coef matrix
% input [nu, mu,z, cf] created by first running
% mfile ellip3d.m or jcoord.m for Johnston atoll
% nu (matrix) and mu (matrix) grid point spacing
% z (matrix) depths cf, center to focus dist in meters.
%  BOUNDARY:
%      bc = 0 is ada = 0,
%      bc = 1 is dn/dmu=0
%      bc = 2 is mixed and based on depth;
%               zbar=mean(mean(z));
%      default bc is 0
%      if bc > 2, apply bc 1 at depths > bc
%                 and bc 0 at depth <= bc
% all measurements in meters
% parameters are: g=9.81 gravity constant
% output is[ada t] where ada represents an JM*IM matrix
% containing the eigen vectors. and t a vector representing
% the eigen modes of oscillation.


cf=cf(1);


% default bc;

  if nargin <5
     bc=0; % apply the boundary condition ada=0
  end

% allow input specified depth to apply boundary conditions
if bc > 2
  zbc = bc;
  bc = 2;

  else
       zbc = mean(mean(z));
```

```
end

zbar = mean(mean(z));


      [IM,JM]=size(mu);

      dnu=abs(nu(2,1)-nu(1,1));

      dmu=abs(mu(1,2)-mu(1,1));

      a=dnu^2;
      b=dmu^2;


% scale depth by the mean depth

z = z/zbar;
zbc=zbc/zbar;

   % verify all depths greater than = 0.0

       for j=1:JM;
          for i=1:IM;
              if z(i,j)<=0.0
                 z(i,j)=0.0;
              end
          end
       end

end


% delete the duplicated coordinate points

   nz=z(1,1:(JM+1)/2-1);

         for i=1:length(nz);

             nz(i)=nan;
```

```
        end

    z(1,1:(JM+1)/2-1)=nz;


    z(IM,(JM+1)/2+1:JM)=nz;

    smu=sinh(mu);

    snu=sin(nu);

    smu=smu.^2;

    snu=snu.^2;

    x=cf/cf*cosh(mu).*cos(nu);

    y=cf/cf*sinh(mu).*sin(nu);


     L=max(max(x))-min(min(x));

% compute hsqd
       hsqd=(smu+snu);
          hsqd(1,(JM+1)/2)=1;
          hsqd(IM,(JM+1)/2)=1;
          hsqd=1./hsqd;


% remove the duplicated points and the singularity
% of hsqd at the foci:

          hsqd(1,1:(JM+1)/2-1)=nz;
          hsqd(IM,(JM+1)/2+1:JM)=nz;
          hsqd(1,(JM+1)/2)=nan;
          hsqd(IM,(JM+1)/2)=nan;



    g = 9.81;

 % matrix of indices
```

```
   I = 1:JM*IM;

   I = reshape(I,JM,IM)';

%%%%%%%% eigenvalue equations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

coef=zeros(IM*JM);

% Interior pts

  for j=2:JM-1
   for  i=3:IM-2

    coef(I(i,j),I(i,j))    = -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
    coef(I(i,j),I(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
    coef(I(i,j),I(i-1,j)) = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));
    coef(I(i,j),I(i,j+1)) = hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
    coef(I(i,j),I(i,j-1)) = hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

   end
  end

i=2;
  for j=2:(JM+1)/2-1

    coef(I(i,j),I(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,JM+1-j)+4*z(i,j));
    coef(I(i,j),I(i-1,JM+1-j)) = hsqd(i,j)*1/(4*a)*(z(i-1,JM+1-j)-z(i+1,j)+4*z(i,j));

  end
i=2;
j=(JM+1)/2;

   coef(I(i,j),I(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
   coef(I(i,j),I(i-1,j+1)) = hsqd(i,j)*1/(4*a)*(z(i-1,j+1)-z(i+1,j)+4*z(i,j));


  for j=(JM+1)/2+1:JM-1

    coef(I(i,j),I(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
    coef(I(i,j),I(i-1,j)) = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

  end
```

```
i=2;

    for j=2:JM-1

      coef(I(i,j),I(i,j))   = -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
      coef(I(i,j),I(i,j+1)) = hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
      coef(I(i,j),I(i,j-1)) = hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

    end

%*********************************************************************

 i=1;

    for j=(JM+1)/2+1:JM-1

        % nu derivitive

    coef(I(i,j),I(i,j))     =   -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
    coef(I(i,j),I(i+1,j))   =    hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i+1,JM+1-j)+4*z(i,j));
    coef(I(i,j),I(i+1,JM+1-j)) = hsqd(i,j)*1/(4*a)*(z(i+1,JM+1-j)-z(i+1,j)+4*z(i,j));

        % mu derivitive

     coef(I(i,j),I(i,j+1)) =     hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
     coef(I(i,j),I(i,j-1)) =     hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

    end

% ***************** focus in rect coord ******************************


dx=abs(x(1,(JM+1)/2+1)-x(2,(JM+1)/2))/2;
dy=abs(y(2,(JM+1)/2-1)-y(2,(JM+1)/2+1))/2;

i=1;

j=(JM+1)/2;

   coef(I(i,j),I(i,j))   = -(2*z(i,j)/dx^2)-(2*z(i,j)/dy^2);
   coef(I(i,j),I(i+1,j)) = 1/(4*dx^2)*(z(i+1,j)-z(i,j+1))+z(i,j)/(dx^2);
   coef(I(i,j),I(i,j+1)) = 1/(4*dx^2)*(z(i,j+1)-z(i+1,j))+z(i,j)/(dx^2);
   coef(I(i,j),I(i+1,j-1)) = 1/(4*dy^2)*(z(i+1,j-1)-z(i+1,j+1))+z(i,j)/(dy^2);
```

```
   coef(l(i,j),l(i+1,j+1)) = 1/(4*dy^2)*(z(i+1,j+1)-z(i+1,j-1))+z(i,j)/(dx^2);


%*************************************************************************

 i=IM-1;

   for j=2:(JM+1)/2-1

       % nu derivitive

    coef(l(i,j),l(i+1,j)) = hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
    coef(l(i,j),l(i-1,j)) = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

  end

  for j=(JM+1)/2+1:JM-1

          % nu derivitive

      coef(l(i,j),l(i-1,j))       = hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,JM+1-j)+4*z(i,j));
      coef(l(i,j),l(i+1,JM+1-j)) = hsqd(i,j)*1/(4*a)*(z(i+1,JM+1-j)-z(i-1,j)+4*z(i,j));

   end

i=IM-1;
j=(JM+1)/2;

% nu derivitive

     coef(l(i,j),l(i+1,j))    =     hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
    coef(l(i,j),l(i-1,j))  =     hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

  for j=2:JM-1

     coef(l(i,j),l(i,j))     =   -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));

      % mu derivitive

     coef(l(i,j),l(i,j+1)) =      hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
     coef(l(i,j),l(i,j-1)) =      hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

  end
```

```
%*************************************************************************

 i=IM;

  for j=2:(JM+1)/2-1            % nu derivitive

    coef(I(i,j),I(i,j))     =   -hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));

    coef(I(i,j),I(i-1,j))   =    hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i-1,JM+1-j)+4*z(i,j));
   coef(I(i,j),I(i-1,JM+1-j))=  hsqd(i,j)*1/(4*a)*(z(i-1,JM+1-j)-z(i-1,j)+4*z(i,j));

        % mu derivitive

     coef(I(i,j),I(i,j+1)) =      hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
    coef(I(i,j),I(i,j-1)) =      hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

  end


%**************** focus in rectangular coordinates **********

i=IM;
j=(JM+1)/2;

   coef(I(i,j),I(i,j))   = -(2*z(i,j)/dx^2)-(2*z(i,j)/dy^2);
   coef(I(i,j),I(i-1,j-1)) = 1/(4*dy^2)*(z(i-1,j-1)-z(i-1,j+1))+z(i,j)/dy^2;
   coef(I(i,j),I(i-1,j+1)) = 1/(4*dy^2)*(z(i-1,j+1)-z(i-1,j-1))+z(i,j)/dy^2;
   coef(I(i,j),I(i-1,j)) = 1/(4*dx^2)*(z(i-1,j)-z(i,j-1))+z(i,j)/dx^2;
   coef(I(i,j),I(i,j-1)) = 1/(4*dx^2)*(z(i,j-1)-z(i-1,j))+z(i,j)/dx^2;


%*************************************************************************

% interior boundary of the island

[n,m]=find(z==0);

 for i=1:length(n)

    if n(i) > 1  & m(i) >1
```

```
        coef(I(n(i),m(i)),I(n(i),m(i)))=0;
        coef(I(n(i),m(i)),I(n(i)-1,m(i)))=0;
        coef(I(n(i),m(i)),I(n(i)+1,m(i)))=0;
        coef(I(n(i),m(i)),I(n(i),m(i)-1))=0;
        coef(I(n(i),m(i)),I(n(i),m(i)+1))=0;

    end
end


%%%%%%%%%%% BOUNDARY CONDITION
****************************************

if bc==1

    j=JM;

  for i=1:IM-1;

    % at mu=MU backward differencing in mu

    coef(I(i,JM),I(i,JM))     =  hsqd(i,j)*3/(2*dmu);
    coef(I(i,JM),I(i,JM-1))   = - hsqd(i,j)*4/(2*dmu);
    coef(I(i,JM),I(i,JM-2))   =   hsqd(i,j)*1/(2*dmu);

   end

       j=1;

   % forward differencing in mu

  for i=2:IM

    coef(I(i,1),I(i,1))     =   - hsqd(i,j)*3/(2*dmu);
    coef(I(i,1),I(i,2))     =   + hsqd(i,j)*4/(2*dmu);
    coef(I(i,1),I(i,3))     =   - hsqd(i,j)*1/(2*dmu);

  end

elseif bc==2
```

```
%****************** mixed derivitve case ******************************************

   o = zbc;  % 'min depth to apply apply dn/dmu=0 ')

% at mu=MU backward differencing in mu

j=JM;
     for i=1:IM-1

        if z(i,j) >= o

           coef(l(i,j),l(i,j))     =   hsqd(i,j)*3/(2*dmu);
           coef(l(i,j),l(i,j-1))   = - hsqd(i,j)*4/(2*dmu);
           coef(l(i,j),l(i,j-2))   =   hsqd(i,j)*1/(2*dmu);

        end
     end



 % forward differencing in mu

j=1;

      for i=2:IM

        if z(i,j) >= o

              coef(l(i,j),l(i,j))    =   - hsqd(i,j)*3/(2*dmu);
              coef(l(i,j),l(i,j+1))  =   + hsqd(i,j)*4/(2*dmu);
              coef(l(i,j),l(i,j+2))  =   - hsqd(i,j)*1/(2*dmu);

        end
      end

end

%[i,j]=find(coef);

%figure;plot(i,j,'.');axis('ij')

%title('coef without deletion of rows')
```

```
% delete unused rows and columns

    f1=l(1,1:(JM+1)/2-1);

    flM=l(lM,(JM+1)/2+1:JM);

        coef(:,flM)=[];
        coef(flM,:)=[];

        coef(:,f1)=[];
        coef(f1,:)=[];

[ii,jj]=find(diag(coef));
coef=coef(ii,ii);

%[i,j]=find(coef);

%figure;plot(i,j,'.');axis('ij')
%   title(' coef matrix with deleted rows')

disp('coef is size')

disp(size(coef))
%keyboard
%cnr=cond(coef);
%disp(['cond = ',num2str(cnr)])

disp('solving coef using matlab eig.m')


[a,l] = eig(coef);  % matlab eigen vector, eigen value function


  %***** unravel lamda squared  and ada **************

    l=abs(diag(l));
    [l,in]=sort(l);

% discard the zero eigenvalues
%    [i,j]=find(l);
%     l=l(i);

% *** sort ada and Lamdasq in decending order
```

```
%
    s=(l*g*zbar)/(cf^2);

    t=(2*pi)^2./s;  % modal period in seconds^2

    t=sqrt(t);

% columns of ada correspond to the sort of Tau

       ada(ii,:)=a(:,:);

       ada=ada(:,in);
```

```
function [ada,t]=eseich3g(nu,mu,z,cf,bc)

% function [ada,t]=eseich3g(nu,mu,z,cf,bc)
% matlab function to determine the 3 dIMensional modes of
% oscillation of an ELLIPTICAL basin under gravity.
%
% calls the mfile eig.m to decompose an JM*IM coef matrix
% input is specifed nu, mu, z, cf created by first running
% mfile rect3d, ellip3d, or jcoord.m
% input required is nu (matrix) and mu (matrix) grid point spacing
% z  (matrix) depths
% cf, center to focus dist in meters.
% BOUNDARY
%       bc = 0 => ada = 0;
%       bc = 1 => dn/dmu =0,
%       bc = 2 = mixed, bc is applied based on depth:
%                   zbar= mean(mean(z));
%       default is bc=0
%       if bc >2, bc 0 is applied at depths < bc
%         and bc 1 is applied at depths >= bc
%
% all measurements in meters
% parameters are: g=9.81 gravity constant
% output is[ada t] where ada represents an JM*IM matrix
% containing the eigen vectors. and t a vector representing
% the eigen modes of oscillation.

disp('eseich3g')
cf=cf(1);
if nargin<5
bc=0; % apply zero bc condition
end

if bc>2

  zbc=bc;
   bc=2;

else

 zbc = mean(mean(z));

end
```

```
zbar = mean(mean(z));

% scale factor for depths

      z=z/zbar;
     zbc=zbc/zbar;


   [IM,JM]=size(mu);

      dnu=abs(nu(2,1)-nu(1,1));

      dmu=abs(mu(1,2)-mu(1,1));

      a=dnu^2;
      b=dmu^2;



% delete the duplicated coordinate points

   nz=z(1,1:(JM+1)/2);

        for i=1:length(nz);

            nz(i)=nan;

        end

    z(1,1:(JM+1)/2)=nz;


    z(IM,(JM+1)/2:JM)=nz;

    smu=sinh(mu);

    snu=sin(nu);

    smu=smu.^2;

    snu=snu.^2;

    x=cf/cf*cosh(mu).*cos(nu);
```

```
    y=cf/cf*sinh(mu).*sin(nu);

      hsqd=smu+snu;

% remove the duplicated points and the singularity of hsqd:

        hsqd(1,1:(JM+1)/2)=nz;
        hsqd(IM,(JM+1)/2:JM)=nz;
        hsqd(1,(JM+1)/2)=nan;
        hsqd(IM,(JM+1)/2)=nan;

        hsqd=1./hsqd;

    g = 9.81;

 % matrix of indices

     I = 1:JM*IM;

     I = reshape(I,JM,IM)';

%%%%%%%% eigenvalue equations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

coef=zeros(IM*JM);

% Interior pts

   for j=2:JM-1
    for  i=3:IM-2

     coef(I(i,j),I(i,j))   =...
-hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
     coef(I(i,j),I(i+1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
     coef(I(i,j),I(i-1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));
     coef(I(i,j),I(i,j+1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
     coef(I(i,j),I(i,j-1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

    end
```

```
   end

i=2;
    for j=2:(JM+1)/2-1

      coef(I(i,j),I(i+1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,JM+1-j)+4*z(i,j));
      coef(I(i,j),I(i-1,JM+1-j)) =...
hsqd(i,j)*1/(4*a)*(z(i-1,JM+1-j)-z(i+1,j)+4*z(i,j));

    end
i=2;

j=(JM+1)/2;

     coef(I(i,j),I(i+1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j+1)+4*z(i,j));
     coef(I(i,j),I(i-1,j+1)) =...
hsqd(i,j)*1/(4*a)*(z(i-1,j+1)-z(i+1,j)+4*z(i,j));


   for j=(JM+1)/2+1:JM-1

      coef(I(i,j),I(i+1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
      coef(I(i,j),I(i-1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

   end

i=2;

    for j=2:JM-1

      coef(I(i,j),I(i,j))   =...
-hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
      coef(I(i,j),I(i,j+1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
      coef(I(i,j),I(i,j-1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

    end
```

```
%*********************************************

 i=1;

    for j=(JM+1)/2+2:JM-1

        % nu derivitive

    coef(I(i,j),I(i,j))      =...
-hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
    coef(I(i,j),I(i+1,j))    =...
hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i+1,JM+1-j)+4*z(i,j));
    coef(I(i,j),I(i+1,JM+1-j)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,JM+1-j)-z(i+1,j)+4*z(i,j));

         % mu derivitive

     coef(I(i,j),I(i,j+1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
     coef(I(i,j),I(i,j-1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

    end

j=(JM+1)/2+1;


    coef(I(i,j),I(i,j+1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i+1,j-1)+4*z(i,j));
     coef(I(i,j),I(i+1,j-1)) =...
hsqd(i,j)*1/(4*b)*(z(i+1,j-1)-z(i,j+1)+4*z(i,j));

    coef(I(i,j),I(i,j))      =...
-hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
    coef(I(i,j),I(i+1,j))    =...
hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i+1,j-2)+4*z(i,j));
    coef(I(i,j),I(i+1,j-2)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,j-2)-z(i+1,j)+4*z(i,j));


% ***** focus in rect coord **************

%dx=abs(x(1,(JM+1)/2+1)-x(2,(JM+1)/2))/2;
```

```
% dy=abs(y(2,(JM+1)/2-1)-y(2,(JM+1)/2+1))/2;

%i=1;

%j=(JM+1)/2;

 % coef(I(i,j),I(i,j))   =...
% -(2*z(i,j)/dx^2)-(2*z(i,j)/dy^2);
 % coef(I(i,j),I(i+1,j)) =...
%1/(4*dx^2)*(z(i+1,j)-z(i,j+1))+z(i,j)/(dx^2);
 % coef(I(i,j),I(i,j+1)) =...
%1/(4*dx^2)*(z(i,j+1)-z(i+1,j))+z(i,j)/(dx^2);
 % coef(I(i,j),I(i+1,j-1)) =...
%1/(4*dy^2)*(z(i+1,j-1)-z(i+1,j+1))+z(i,j)/(dy^2);
 % coef(I(i,j),I(i+1,j+1)) =...
%1/(4*dy^2)*(z(i+1,j+1)-z(i+1,j-1))+z(i,j)/(dx^2);


%*************************************

 i=IM-1;

   for j=2:(JM+1)/2-1

       % nu derivitive

    coef(I(i,j),I(i+1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,j)-z(i-1,j)+4*z(i,j));
    coef(I(i,j),I(i-1,j)) =...
hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j)+4*z(i,j));

  end

  for j=(JM+1)/2+1:JM-1

          % nu derivitive

      coef(I(i,j),I(i-1,j))   =...
hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,JM+1-j)+4*z(i,j));
      coef(I(i,j),I(i+1,JM+1-j)) =...
hsqd(i,j)*1/(4*a)*(z(i+1,JM+1-j)-z(i-1,j)+4*z(i,j));

   end
```

```
i=IM-1;
j=(JM+1)/2;

% nu derivitive

     coef(I(i,j),I(i+1,j-1))    =...
hsqd(i,j)*1/(4*a)*(z(i+1,j-1)-z(i-1,j)+4*z(i,j));
     coef(I(i,j),I(i-1,j))  =...
hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i+1,j-1)+4*z(i,j));

  for j=2:JM-1

     coef(I(i,j),I(i,j))     =...
-hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));

      % mu derivitive

     coef(I(i,j),I(i,j+1)) = ...
hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
     coef(I(i,j),I(i,j-1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

  end


%*****************************************

 i=IM;

  for j=2:(JM+1)/2-2            % nu derivitive

    coef(I(i,j),I(i,j))     =...
-hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));

    coef(I(i,j),I(i-1,j))   =...
hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i-1,JM+1-j)+4*z(i,j));
   coef(I(i,j),I(i-1,JM+1-j))=...
hsqd(i,j)*1/(4*a)*(z(i-1,JM+1-j)-z(i-1,j)+4*z(i,j));

        % mu derivitive

     coef(I(i,j),I(i,j+1)) =....
hsqd(i,j)*1/(4*b)*(z(i,j+1)-z(i,j-1)+4*z(i,j));
```

```
    coef(I(i,j),I(i,j-1)) = ...
hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i,j+1)+4*z(i,j));

  end

j=(JM+1)/2-1;


    coef(I(i,j),I(i-1,j)) = ...
hsqd(i,j)*1/(4*a)*(z(i-1,j)-z(i-1,j+2)+4*z(i,j));
     coef(I(i,j),I(i-1,j+2)) = ...
hsqd(i,j)*1/(4*a)*(z(i-1,j+2)-z(i-1,j)+4*z(i,j));

    coef(I(i,j),I(i,j))      = ....
-hsqd(i,j)*((2*z(i,j)/a)+(2*z(i,j)/b));
    coef(I(i,j),I(i-1,j+1))   =...
hsqd(i,j)*1/(4*b)*(z(i-1,j+1)-z(i,j-1)+4*z(i,j));
    coef(I(i,j),I(i,j-1)) =...
hsqd(i,j)*1/(4*b)*(z(i,j-1)-z(i-1,j+1)+4*z(i,j));



%******* focus in rectangular coordinates ********

%i=IM;
%j=(JM+1)/2;

%   coef(I(i,j),I(i,j))   = -(2*z(i,j)/dx^2)-(2*z(i,j)/dy^2);
%   coef(I(i,j),I(i-1,j-1)) =... %
%1/(4*dy^2)*(z(i-1,j-1)-z(i-1,j+1))+z(i,j)/dy^2;
%   coef(I(i,j),I(i-1,j+1)) =
%1/(4*dy^2)*(z(i-1,j+1)-z(i-1,j-1))+z(i,j)/dy^2;
%   coef(I(i,j),I(i-1,j)) =
%1/(4*dx^2)*(z(i-1,j)-z(i,j-1))+z(i,j)/dx^2;
%   coef(I(i,j),I(i,j-1)) =
%1/(4*dx^2)*(z(i,j-1)-z(i-1,j))+z(i,j)/dx^2;


%****** interior boundary of the island *******

[n,m]=find(z==0);

 for i=1:length(n)
```

```
    if n(i) > 1 & m(i) > 1


        coef(l(n(i),m(i)),l(n(i),m(i)))=0;
        coef(l(n(i),m(i)),l(n(i)-1,m(i)))=0;
        coef(l(n(i),m(i)),l(n(i)+1,m(i)))=0;
        coef(l(n(i),m(i)),l(n(i),m(i)-1))=0;
        coef(l(n(i),m(i)),l(n(i),m(i)+1))=0;

     end
 end



%%%%%%%%%%% BOUNDARY CONDITION %%%%%%%%%%%%%%%

if bc==1

      j=JM;

   for i=1:IM-1;

      % at mu=MU backward differencing in mu

      coef(l(i,JM),l(i,JM))      =  hsqd(i,j)*3/(2*dmu);
      coef(l(i,JM),l(i,JM-1))    = - hsqd(i,j)*4/(2*dmu);
      coef(l(i,JM),l(i,JM-2))    =   hsqd(i,j)*1/(2*dmu);

    end

         j=1;

    % forward differencing in mu

   for i=2:IM

      coef(l(i,1),l(i,1))      =   - hsqd(i,j)*3/(2*dmu);
      coef(l(i,1),l(i,2))      =   + hsqd(i,j)*4/(2*dmu);
      coef(l(i,1),l(i,3))      =   - hsqd(i,j)*1/(2*dmu);

   end

elseif bc==2
```

```
%****************** mixed derivitve case
%*****************************************


   o = zbc;  % 'min depth to apply apply dn/dmu=0 ')

% at mu=MU backward differencing in mu

j=JM;
     for i=1:IM-1

        if z(i,j) >= o

           coef(I(i,j),I(i,j))     =   hsqd(i,j)*3/(2*dmu);
           coef(I(i,j),I(i,j-1))   = - hsqd(i,j)*4/(2*dmu);
           coef(I(i,j),I(i,j-2))   =   hsqd(i,j)*1/(2*dmu);

        end
     end



 % forward differencing in mu

j=1;

       for i=2:IM

         if z(i,j) >= o

                coef(I(i,j),I(i,j))    =...
 -hsqd(i,j)*3/(2*dmu);
                coef(I(i,j),I(i,j+1))  =...
 +hsqd(i,j)*4/(2*dmu);
                coef(I(i,j),I(i,j+2))  =...
  -hsqd(i,j)*1/(2*dmu);

         end
      end

end
```

```
% [i,j]=find(coef);

% figure;plot(i,j,'.');axis('ij')

% title('coef without deletion of rows')

% delete unused rows and columns

    f1=l(1,1:(JM+1)/2);

    flM=l(IM,(JM+1)/2:JM);

        coef(:,flM)=[];
        coef(flM,:)=[];

        coef(:,f1)=[];
        coef(f1,:)=[];



[ii,jj]=find(diag(coef));
coef=coef(ii,ii);

% [i,j]=find(coef);

%figure;plot(i,j,'.');axis('ij')
 %  title(' coef matrix with deleted rows')

disp('coef is size')

disp(size(coef))
%keyboard
%cnr=cond(coef);
%disp(['cond = ',num2str(cnr)])

disp('solving coef using matlab eig.m')


[a,l] = eig(coef);  % matlab eigen vector, eigen value function


  %***** unravel lamda squared  and ada **************
```

```
    l=abs(diag(l));
    [l,in]=sort(l);

% discard the zero eigenvalues

 % [i,j]=find(l);
 %    l=l(i);

% *** sort ada and Lamdasq in decending order
%
    s=(l*g*zbar)/(cf^2);

    t=(2*pi)^2./s;  % modal period in seconds^2
    t=sqrt(t);


% columns of ada correspond to the sort of Tau

      ada(ii,:)=a(:,:);

     ada=ada(:,in);
```

```
function [lon,lat]=ij2ltln(n,m,dx,dy);
%function [lon,lat]=ij2ltln(n,m,dx,dy);
% makes an [i,j] grid and converts it to long lat
% at 16 geg n lat
% xo,yo are lon,lat at pt (1,1)
% specify starting grid point and resolution in meters
% input [n,m] is size of desired output matrices
% where n represents y coord (lat)
% and m represents x coord (lon)
% output is two vectors or matrices
% lat is a row matrix of constant columns and,
% lon is column matrix of constant rows,
% as in meshgrid, converted to lat/lon,

% for jb250m.dat;
% Modify this line for other lat/lon:
 xo=-169-35.95/60;
 yo=16+38.3/60;
lat=1:n;
lon=1:m;
[lon,lat]=meshgrid(lon,lat);
% meters to lat/lon conversion:
% 1 deg N lat = 110.65 km, 1 deg lon = 107.04 km
% at 16 deg N.
% modify this for other lat/lon:
lat=yo+((dy/(1000*110.65))*lat);
```

```
function [NU,MU,xp1,yp1,Z,Cf]=jcoord(dx,dy,bathy)

% function [NU,MU,xp1,yp1,Z,Cf]=jcoord(dx,dy,bathy)
%
% interpolates and tranlates a subset of
% the input matrix (bathy) into  elliptical cooordintes
% dx and dy are the resolution of bathy in meters
% transforms cartesian x, y coord to elliptical cylindrical coord
% involves rotataion, translation and interpolation
% output is, NU,MU coordinates of  bay
% bathymetry, xp1,yp1, geographical coordinates of NU , MU
% interpolated bathymetry Z ,
% and  Cf; the center to focus distance, semi-major axis,
% and semi-minor axis
%  for use in esiech3g.m or esiech3f.m
% this function expects 2 functions to be in the matlabpath:
%                    calls ij2latln.m and xy2latlon.m
% for coordinate transformations to latitude and longitude
 %

z=bathy;

[IM,JM]=size(z)

 x=0:dx:(JM-1)*dx;
 y=0:dy:(IM-1)*dy;

[x,y]=meshgrid(x,y);

contour(x,y,z,[0.1,6,10,20])

   axis('equal')
   hold on
   grid on

disp('choose long axis end points')
[x1,y1]=ginput(2)
plot(x1,y1,'w')
disp('choose short axis end points')
[x2,y2]=ginput(2)

   plot(x2,y2,'w')
```

```
L=sqrt((x1(1)-x1(2))^2 + (y1(1)-y1(2))^2)

W=sqrt((x2(1)-x2(2))^2 + (y2(1)-y2(2))^2)

% compute semi-major and semi-minor axes

A=L/2
B=W/2

% compute center to focus distance

Cf=sqrt(A^2-B^2);

% if d==1

% Cf in meters

% A=L*sqrt((1000*110.65)^2+(1000*107.04)^2)/2;
% B=W*sqrt((1000*110.65)^2+(1000*107.04)^2)/2;
% Cf=sqrt(A^2-B^2);


end
% compute Max value of Mu and compute nu, mu coord matrices

Mu=asinh(B/Cf)

res=input('specify your resolution [Nu(IM rows),mu(JM columns)] even integers
');

dnu=pi/res(1);
Nu=pi;
dmu=Mu/res(2);
u=0:dmu:Mu;
u=[-fliplr(u(2:length(u))),u];

% (columns of equal radius);

IM=length(u);
nu=-Nu:dnu:0;

%  (rows of equal radius)
```

```
JM=length(nu);

[MU,NU]=meshgrid(u,nu);

xp =Cf*cosh(MU).*cos(NU);
yp =Cf*sinh(MU).*sin(NU);

    %plot(xp,yp,'.')

disp('computing rotation angle through long axix of the bay')

     alpha=atan((y1(2)-y1(1))/(x1(2)-x1(1)));

disp('choose xo,yo offset')

    [xo,yo]=ginput(1)

    plot(xo,yo,'y*')

% transform the coordinates

 xp1=xp*cos(alpha)-yp*sin(alpha)+xo(1);
 yp1=xp*sin(alpha)+yp*cos(alpha)+yo(1);


%xx=(x-xo)*cos(alpha)+(y-yo)*sin(alpha);
%yy=(y-yo)*cos(alpha)-(x-xo)*sin(alpha);

 plot(xp1,yp1,'.')

%keyboard

Z=interp2(x,y,z,xp1,yp1);

figure
    mesh(xp1,yp1,-Z);axis('equal')
    title('Interpolated Bathymetry')

%figure; plot(xp1,yp1,'.')
% hold on
% plot3(xp1,yp1,Zl,'*')
Cf=[Cf,A,B];
```

```
function [ad1,ADII] = jplot3ef(xpl,ypl,ada,t)

%function [ad1,ADII] = jplot3ef(xpl,ypl,ada,t)
% meshplots output from eseiche3f.m 3d representation of
% of Johnston Atoll
% uses (xpl,ypl,ada,t) generated from eseich3f.m
% expects the files jb250m.dat, calls ij2latln.m
% to be in the matlabpath
% contours ada after tranforming to an xy uniform
% grid
disp('jplot3ef.m')

load jb250m.dat
  [n,m]=size(jb250m);
  dx=250;
  dy=250;

deg=menu('Is x,y in degrees ?','Yes','No')

if deg==1

   [lon,lat]=ij2latln(n,m,dx,dy);
else

  lon=0:dx:(m-1)*dx;
  lat=0:dy:(n-1)*dy;
  [lon,lat]=meshgrid(lon,lat);

end

[IM,JM]=size(xpl);  % size of matrix


mode=input('mode to plot?');

ad=ada(:,mode);

I=1:IM*JM;

II=reshape(I,JM,IM)';


    f1 = II(1,1:(JM+1)/2-1)
```

```
    flM = ll(lM,(JM+1)/2+1:JM)

  l(flM) = [];

   l(f1) = [];

l=l';


ad1(l)=ad;

AD1=ad;

 x=xpl'; y=ypl';

 x=reshape(x,lM*JM,1);
 y=reshape(y,lM*JM,1);


x(flM)=[];
x(f1)=[];

y(flM)=[];
 y(f1)=[];




% get rid of duplicated points


ad1(length(ad1)+1:lM*JM)=zeros(size(flM));

ad1=reshape(ad1,JM,lM)';


   nn=zeros(size(f1));

     for i=1:length(nn)

         nn(i)=nan;
```

```
    end


ad1(1,1:(JM+1)/2-1)= fliplr(ad1(1,(JM+1)/2+1:JM));
ad1(IM,(JM+1)/2+1:JM)=fliplr(ad1(IM,1:(JM+1)/2-1));
%ad1(1,(JM+1)/2)=nan;
%ad1(IM,(JM+1)/2)=nan;


figure;

    mesh(xpl,ypl,ad1);axis('equal');
    title(['Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
    ' (min)  ellip Basin, bc= ',num2str(bc)])
    xlabel('non dimensional distance ')
    ylabel('nondimensional distance ')

% this part contours eta by transforming
% x,y,eta to an evenly spaced grid
% figure out a way to contour Eta
% from ellip3b



     XPo=min(x);
   Xpmax=max(x);
     dxp=(Xpmax-XPo)/(15);
     xpp=XPo:dxp:Xpmax;
     YPo=min(y);
   Ypmax=max(y);
     dyp=dxp; % (Ypmax-YPo)/(10);
     ypp=YPo:dyp:Ypmax;


[XPP,YPP]=meshgrid(xpp,ypp); % uniform grid


    ADII=griddata(x,y,AD1,XPP,YPP);



% contour vector
     dv=(max(max(AD1))-min(min(AD1)))/5;
```

```
    vo=min(min(AD1));
    vmax=max(max(AD1));
    v=vo:dv:vmax;


figure;
    plot(xpl(:,1),ypl(:,1),'w',xpl(:,JM),ypl(:,JM),'w');

    hold on

    c=contour(XPP,YPP,ADII,[0,v],'r--');axis('equal');
    clabel(c,'manual')
    title(['Contours of Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
      ' (min)  ellip Basin, bc= ',num2str(bc)])
    xlabel('Longitude')
    ylabel('Lattitude')

    c=contour(lon,lat,jb250m,[0.1,20],'y');
      axis('equal')

hold off
```

```
function [ad1,ADII] =jplot3eg(xpl,ypl,ada,t)

%function [ad1,ADII] = jplot3eg(xpl,ypl,ada,t)
%
% meshplots output from ellip3g.m, 3d representation
% of seiche modes
% SPECIFICALLY FOR JOHNSTON ATOLL
% input is xpl, ypl from ellip3d.m or jcoord.m
% Uses (a,t) generated from eseich3g.m
% Expects the files jb250m.dat, and ij2latln.m
% to be in the matlabpath
% Contours ada after tranforming to an xy uniform
% grid

disp('jplot3eg.m')

load jb250m.dat
  [n,m]=size(jb250m);
  dx=250;
  dy=250;

deg=menu('are xpl,ypl in degrees ?','Yes','No','Convert to deg')

if deg==1

   [lon,lat]=ij2latln(n,m,dx,dy);
    mlon=xpl;
    mlat=ypl;

    xaxl='Degrees Longitude';
    yaxl='Degrees Latitude';

elseif deg==2

  lon=0:dx:(m-1)*dx;
  lat=0:dy:(n-1)*dy;
  [lon,lat]=meshgrid(lon,lat);
   mlon=xpl;
   mlat=ypl;

    xaxl='Meters';
    yaxl='Meters';
```

```
elseif deg==3

   [mlon,mlat]=xy2latln(xpl,ypl);
   [lon,lat]=ij2latln(n,m,dx,dy);
   xaxl='Degrees Longitude';
    yaxl='Degrees Latitude';

end
end

[IM,JM]=size(mlon);  % size of the reformed matrix

mode=input('mode to plot?');

ad=ada(:,mode);

l=1:IM*JM;

ll=reshape(l,JM,IM)';


    f1 = ll(1,1:(JM+1)/2);

    flM = ll(IM,(JM+1)/2:JM);

  l(flM) = [];

   l(f1) = [];

l=l';

ad1(l)=ad;
AD1=ad;

 x=mlon'; y=mlat';

 x=reshape(x,IM*JM,1);
 y=reshape(y,IM*JM,1);


x(flM)=[];
x(f1)=[];
```

```
y(flM)=[];
 y(f1)=[];



% get rid of duplicated points


ad1(length(ad1)+1:IM*JM)=zeros(size(flM));

ad1=reshape(ad1,JM,IM)';


   nn=zeros(size(f1));

     for i=1:length(nn)

         nn(i)=nan;
     end

ad1(1,1:(JM+1)/2)= fliplr(ad1(1,(JM+1)/2:JM));
ad1(IM,(JM+1)/2:JM)=fliplr(ad1(IM,1:(JM+1)/2));
ad1(1,(JM+1)/2)=nan;
ad1(IM,(JM+1)/2)=nan;


ms=menu('mesh plot of the mode ?','Yes','No');
if ms==1


   mesh(mlon,mlat,ad1);axis('equal');
   title(['Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
   ' (min)'])
   xlabel(xaxl)
   ylabel(yaxl)
   zlabel('Nondimensional Elevation')

end
keyboard

% this part contours eta by transforming
```

```
% x,y,eta to an evenly spaced grid
% figure out a way to contour Eta
% from ellip3b


    XPo=min(x);
  Xpmax=max(x);
    dxp=(Xpmax-XPo)/(15);
    xpp=XPo:dxp:Xpmax;
    YPo=min(y);
  Ypmax=max(y);
    dyp=dxp; % (Ypmax-YPo)/(10);
    ypp=YPo:dyp:Ypmax;


[XPP,YPP]=meshgrid(xpp,ypp); % uniform grid


    ADII=griddata(x,y,AD1,XPP,YPP);


% contour vector
     dv=(max(max(AD1))-min(min(AD1)))/6;
     vo=min(min(AD1));
     vmax=max(max(AD1));
     v=vo+dv:dv:vmax;

    v=[0,vo,v,vmax];

ms=menu('contour plot of the mode ?','Yes','No')

if ms==1

    plot(mlon(:,1),mlat(:,1),'w',mlon(:,JM),mlat(:,JM),'w');

    hold on

    c=contour(XPP,YPP,ADII,[0,v],'r--');axis('equal');
    clabel(c,'manual')
    title(['Contours of Mode ',num2str(mode),' T= ',num2str(t(mode)/60),...
      ' (min)'])
    %xlabel('Longitude')
    %ylabel('Lattitude')
```

```
   xlabel(xaxl)
   ylabel(yaxl)
%   zlabel('Nondimensional Elevation')

     c=contour(lon,lat,jb250m,[0.1,20],'y');
       axis('equal')

hold off
end
```

```
function [x,y,z]=rect2d(L,W,Hmax,Hmin,dx);

%function [x,y,z]=rect2d(L,W,Hmax,Hmin,dx);
% L=length,W=width,Hmax= max depth Hmin = min depth
% if Hmax=Hmin, z is uniform depth;
% otherwise bay is uniformly sloping from Hmax to Hmin;
% forms a rectangular basin for use in seiche2d.m
%
% if dx not specified, default is 250 meters
if nargin <=4

dx=250;
else
end
x=0:dx:L;
x=x';
n=length(x);

if Hmax==Hmin;

 z=zeros(n,1);
 z=z+H;

else

dz=(Hmax-Hmin)/(n-1)
z=Hmax:-dz:Hmin;
z=z';
end

y=zeros(n,1);
y=y + W;
```

```
function [x,y,z]=rect3d(L,W,dx,dy,Hmax,Hmin)
%function [x,y,z]=rect3d(L,W,dx,dy,Hmax,Hmin)
% generates a basin to run rseich3d.m
% L= Length of basin
% W = Width of basin
% dx, dy are interval spacings where L=N*dx, W=M*dy
%
% Hmax, Hmin are max and min bay depths
% the east and west boundaries are depth (Hmax-Hmin)/2
 x=0:dx:L;
 y=0:dy:W;

[x,y]=meshgrid(x,y);
[IM,JM]=size(x);
z=zeros(size(x));

if Hmax==Hmin
   z=z+Hmax
 else
    dz=(Hmax-Hmin)/(IM-1)
    zb=Hmax:-dz:Hmin;
    zb=zb';
    for j=1:JM
    z(:,j)=z(:,j)+zb;
end
end
% z(:,1)=zeros(IM,1)+mean(zb);
% z(:,JM)=z(:,1);
disp('basin created')
disp('x  y  z are size: ')
disp(size(x))
```

```
function [ada,t]=rseich3d(x,y,z,bc)
%
% function [ada,t]=rseich3d(x,y,z,bc)
% can be called from RECT3d.m for std rect bathy
% matlab mfile to determine the 3 dimensional modes of
% oscillation of a semi-inclosed rectangular bay under gravity.
%  calls the mfile eig.m to decompose an N*N coef matrix
% input required is x,y,z (matrices)
% as from rect3d.m or in the format as from 'meshgrid'
% BOUNDARY
%        bc = 0, ada =0;
%        bc = 1, dada/dx, dada/dy = 0
%        bc = 2,  apply bc based on depth: zbar = mean(mean(z))
%        if bc >2, apply bc based on depth = bc
%        default bc is bc = 0
% all measurements in meters
% parameters are: g=9.81 gravity constant
% output is[ada Tau] where ada represents a matrix
% containing the eigenvectors. and t a vector representing
% the eigenvalues, modes of oscillation
% OUTPUT PLOTTED with 'aplot3r.m
%
if nargin < 4
   bc=0;
end

if bc>2

  zbc=bc;

   bc=2; % apply boundary as function of depth

  else

zbc=mean(mean(z));

end

% depth scale factor

zbar=mean(mean(z));

[IM,JM]=size(z);
```

```
g=9.81;
L1=max(max(x))-min(min(x));
L=L1/L1;
dx=L/JM;
W1=max(max(y))-min(min(y));
W=W1/L1;
dy=W/IM;

 % *********** INDEX MATRIX ************************

I= 1:IM*JM;
I = reshape(I,JM,IM)';
coef=zeros(IM*JM);

% scale factors

a=dy^2;
b=dx^2;

 for  i=2:IM-1
  for j=2:JM-1

   coef(I(i,j),I(i,j))   = -(2*z(i,j)/b)-(2*z(i,j)/a);
   coef(I(i,j),I(i+1,j)) = 1/(4*a)*(z(i+1,j)-z(i-1,j))+z(i,j)/a;
   coef(I(i,j),I(i-1,j)) = 1/(4*a)*(z(i-1,j)-z(i+1,j))+z(i,j)/a;
   coef(I(i,j),I(i,j+1)) = 1/(4*b)*(z(i,j+1)-z(i,j-1))+z(i,j)/b;
   coef(I(i,j),I(i,j-1)) = 1/(4*b)*(z(i,j-1)-z(i,j+1))+z(i,j)/b;

end
end

% X=0 boundary forward dif in y
if bc==1
 i=1;
 for j=2:JM-1;

   coef(I(i,j),I(i,j))   = -3/(2*dy);
   coef(I(i,j),I(i+1,j)) =  4/(2*dy);
   coef(I(i,j),I(i+2,j)) = -1/(2*dy);

 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
 i=IM;
 for j=2:JM-1;
 % at X=M backward differencing in y

   coef(I(i,j),I(i,j))   = +3/(2*dy);
   coef(I(i,j),I(i-1,j)) = -4/(2*dy);
   coef(I(i,j),I(i-2,j)) =  1/(2*dy);

 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
 % at y=0 forward differencing in x
 j=1;
 for i=1:IM;

   coef(I(i,j),I(i,j))   = -3/(2*dx);
   coef(I(i,j),I(i,j+1)) =  4/(2*dx);
   coef(I(i,j),I(i,j+2)) = -1/(2*dx);

 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55
 % at x = L backward differencing in x
 j=JM;
 for i=1:IM

   coef(I(i,j),I(i,j))   = +3/(2*dx);
   coef(I(i,j),I(i,j-1)) = -4/(2*dx);
   coef(I(i,j),I(i,j-1)) =  1/(2*dx);

 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%

elseif bc==2 % mixed bdry

% apply bc based on depth zbc;

% at x=L backward difference in x

j=JM;
 for i=1:IM
```

```
  if z(i,j)>zbc

   coef(I(i,j),I(i,j))   = +3/(2*dx);
   coef(I(i,j),I(i,j-1)) = -4/(2*dx);
   coef(I(i,j),I(i,j-1)) =  1/(2*dx);

 end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

 % at y=0 forward differencing in x

 j=1;
 for i=1:IM;
if z(i,j)>zbc;

   coef(I(i,j),I(i,j))   = -3/(2*dx);
   coef(I(i,j),I(i,j+1)) =  4/(2*dx);
   coef(I(i,j),I(i,j+2)) = -1/(2*dx);

 end
end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% at y= 0 forwards diff in y

 i=1;
 for j=2:JM-1;

 % at X=M forwards differencing in y
   if z(i,j)>zbc

   coef(I(i,j),I(i,j))   = -3/(2*dy);
   coef(I(i,j),I(i+1,j)) = +4/(2*dy);
   coef(I(i,j),I(i+2,j)) =  -1/(2*dy);

 end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

end
```

```
disp('solving coef using MATLAB eig.m')

   [ada,l] = eig(coef);  % matlab eigen vector, eigen value function
   l = abs(diag(l));  % only keep the non-zero values of lamdalam=abs(lam);

%***** unravel lamda squared  and ada **************

   [l,in]=sort(l);

% *** sort ada and Lamdasq in decending order

   s=(l*g*zbar/(L1^2));
   t=(2*pi)^2./s;
   t=sqrt(t);  % modal period in in seconds

% columns of ada correspond to the sort of Tau

   ada=ada(:,in);
   [dd,ee]=find(t<inf);
   t=t(dd);
   ada=ada(:,dd);
```

```
function [a,t]=seiche2d(x,b,h,c)

% function [a,t]=seiche2d(x,b,h,c)
% matlab mfile to determine the 2 dimensionsl modes of
% oscillation of a bay under gravity
%  calls the mfile eig.m to decompose an N*N coef matrix
% input required is x(vextor),h=depths(vector),b=widths(vector)
% all of length (IM).
% boundary condition (bc) specified as:
%           h==>0 at x(IM)
%           or h is finite at x(IM), bc is automatically applied
% conditional upon h(IM)< mean(h)
% for i=1:IM
% x(i) is distance along the talweg: represents the length
% of the bay
% x(i) must increase monotonically
% h(i) is the crossectional average depth at x(i)
% b(i) is the width at x(i)
% normalization: bbar(i)=b(i)/max(x)
% hbar(i)=h(i)/zbar
% zbar is depth at x(0)
% all measurements in meters
% parameter: g = 9.81 (m/s^2) gravity constant
% output is [a t] where a is an N*N matrix
% of eigen vectors (modes), and t is a 1*N  vector representing
% the eigen values (periods of oscillation in seconds)
% c is a code, if c=1 a plot of the coef matrix is
% ploted
% otherwise no plot is given
nna=nargin;
if nargin < 4
  c=0;
else
c=1;
end

g=9.81;
L=max(x)-min(x);
zbar=h(1);
[IM]=length(x);
dx=L/(IM-1)
bbar=b/L;
hbar = h./zbar;
```

```
end

  %main diagonal
coef=zeros(IM,IM);
for i=1:IM
 coef(i,i)=2*hbar(i);
end
for i=2:IM-1
 coef(i,i+1)= -(hbar(i)+(1/4)*(hbar(i+1)-hbar(i-1)+ ...
       (hbar(i)/bbar(i))*(bbar(i+1)-bbar(i-1))));
end
for i=2:IM-1
 coef(i,i-1)=-(hbar(i)-(1/4)*(hbar(i+1)-hbar(i-1)+ ...
       (hbar(i)/bbar(i))*(bbar(i+1)-bbar(i-1))));
end


% if h(IM) aproaches zero

 if h(IM) < mean(h)

     coef(IM,IM)   =  3/4*(4*hbar(IM-1)-hbar(IM-2));
     coef(IM,IM-1) = -1*(4*hbar(IM-1)-hbar(IM-2));
     coef(IM,IM-2) = 1/4*(4*hbar(IM-1)-hbar(IM-2));
 disp('h ==> 0 , bc=0')
 else

  % h is a finite depth at N

        coef(IM,IM)    = +2*hbar(IM);
        coef(IM,IM-1)  = -5*hbar(IM);
        coef(IM,IM-2)  = +4*hbar(IM);
        coef(IM,IM-3)  = -hbar(IM);

disp('h is finite, bc==1')
  end
end
if c==1
[i,j]=find(coef);
plot(i,j,'.'); axis('ij');axis('equal')
title('2 Dimensional coef Matrix')
end
```

```
 [a l] = eig(coef);

% matlab eigen vector, eigen value function

 l = abs(diag(l));
[l,in]=sort(l);

%***** unravel lamda squared **************

sigmasq=(l*IM^2*g*zbar)/(L^2);
sigma=sqrt(sigmasq);
t=(2*pi)./sigma;

 % modal period in seconds
% *** sort ada and Tau in decending order

a=a(:,in);

  % columns of a correspond to the sort of t
```

```
function [lon,lat]=xy2ltln(x,y);
%function [lon,lat]=xy2ltln(x,y);
% takes coordinate vectors or matrices
% in meters and and converts them to long lat
% at 16.4 deg n lat
% input [x,y] are matrices of x, and y coordinate pairs: distances in meters.
% x and y need not be evenly spaced, as from meshgrid.
% conversion to lat / lon :
%       where  y coord => (lat)
%       and  x coord => (lon)
% output is two vectors or matrices [lon,lat] of size(x).
% if x and y are the output from mesgrid, then
% lat is a row matrix of constant columns and,
% lon is column matrix of constant rows,
% converted to lat/lon,

% for jb250m.dat;
 xo=-169-35.95/60;
 yo=16+38.3/60;


lat=yo+((1/(1000*110.65)).*y);
lon=xo+((1/(1000*107.04)).*x);
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5101	2
3.	Chairman (Code OC/Co) Department of Oceanography Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor C. A. Collins (OC/Co) Department of Oceanography Naval Postgraduate School Monterey, CA 93943 - 5000	1
5.	Professor Everett Carter (OC/Cr) Department of Oceanography Naval Postgraduate School Monterey, CA 93943 - 5000	1
6.	Professor Edward B. Thornton (OC/Tm) Department of Oceanography Naval Postgraduate School Monterey, CA 93943 - 5000	1
7.	LCDR Burton T. Palmer 6809 Clara Lee Avenue San Diego, CA 92120	1
8.	Dr. Phillip Lobel WoodsHole Oceanographic Institution Woods Hole, MA 02543	1

9. Dr. Thomas Kinder — 1
Office of Naval Research (Code 322 CD)
Physical Oceanography
800 N. Quincey
Arlington, VA 22217 - 5000

10. Mike Cook (OC/Ck) — 1
Department of Oceanography
Naval Postgraduate School
Monterey, CA 93943 - 5000

11. Dr. Joseph J. Cooney — 1
Environmental Sciences Program
University of Massachusetts, Boston
100 Morrisey Boulevard
Boston, MA 02125-3393

12. Dr. Roger Lukas — 1
Department of Oceanography
University of Hawaii
Honolulu, HI 96822

PROGRAM AT RESERVE DESK